

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M 2612 – Elektrotechnika a informatika

Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika

Návrh dvouosého lineárního servopohonu

Design biaxial linear servo-drive

Diplomová práce

Autor: **Jan Zelenka**

Vedoucí diplomové práce: Doc. Ing. Petr Tůma, CSc.

Konzultant: Ing. Jaroslav Vlach, Ing. Martin Šolc (Preciosa,
Jablonec nad Nisou)

Liberec 15.5.2008

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé diplomové práce Doc. Ing. Petru Tůmovi, CSc. za cenné připomínky k mé diplomové práci. Dále konzultantům Ing. Jaroslavu Vlachovi a Ing. Martinu Šolcovi za konzultace odborných aspektů diplomové práce. Nakonec bych chtěl poděkovat svým rodičům za podporu po celou dobu studia.

Abstrakt

Diplomová práce popisuje vlastnosti krokových motorů, lineárních motorů a pneumatického servopohonu, které by mohly být využity jako technické prostředky pro realizaci dvouosého lineárního servopohonu. Dále jsou v práci popsány vlastnosti a komunikace řídicí jednotky MR-MG30 od firmy Mitsubishi Electric.

V další části práce je popsán model dvouosého lineárního servopohonu, ve kterém je využita řídicí jednotka MR-MG30. Práce se zde také zabývá tvorbou a funkcemi řídicího programu pro PLC, které prostřednictvím jednotky MR-MG30 celý model servopohonu ovládá. Řídicí program pro PLC byl vytvořen v prostředí GX IEC Developer 7.00 a je realizován jako čtveřice funkčních bloků (podprogramů). Model servopohonu je obsluhován pomocí ovládacího panelu PLC, pro který byl také napsán obslužný program. Program pro ovládací panel byl napsán v software E-Designer 7. Program pro ovládací panel je složen ze čtyř programových bloků, které reprezentují čtyři obrazovky PLC.

Klíčová slova: MR-MG30, Mitsubishi Electric, GX IEC Developer 7.00, E-Designer 7

Abstract

This diploma thesis describes properties of stepper motors, linear motors and pneumatic servo-drives which may be used as technical means of implementation of biaxial linear servo-drive. Properties and communication possibilities of MR-MG30 control unit by Mitsubishi Electric are also discussed.

Model of biaxial linear servo-drive with MR-MG30 control unit is described in the next part of the thesis as well as creation of control software for PLC controlling the whole model with use of the MR-MG30 unit. The control software was created using GX IEC Developer 7.00 suite and it is composed of four function blocks (subroutines). The model of the servo-drive is operated from control panel of the PLC. Software for this panel was also created, with use of E-Designer 7 software. Software for control panel consists of four program blocks representing four PLC screens.

Keywords: MR-MG30, Mitsubishi Electric, GX IEC Developer 7.00, E-Designer 7

Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
Obsah.....	6
Seznam ilustrací.....	7
Seznam použitých termínů a zkratek.....	9
Úvod.....	11
1 Krokové motory.....	12
1.1 Funkce krokových motorů.....	12
1.1.1 Krokový motor s proměnným magnetickým odporem.....	13
1.1.2 Krokový motor s permanentním magnetem.....	14
1.2 Řízení krokových motorů.....	16
1.3 Mikrokrokování.....	20
2 Lineární motory.....	22
2.1 Konstrukce lineárního motoru.....	22
2.2 Synchronní lineární motor.....	23
2.3 Asynchronní lineární motor.....	23
2.4 Tubulární lineární motor.....	24
2.5 Krokový lineární motor.....	24
2.6 „Bezželezný“ lineární motor.....	26
2.7 Příslušenství lineárních motorů.....	26
3 Pneumatický servopohon.....	28
3.1 Pneumatický válec.....	28
3.2 Pneumatický servoventil.....	29
4 Řídicí jednotka MR-MG30.....	30
4.1 Funkce a parametry MR-MG30.....	30
4.2 PROFIBUS-DP.....	32
4.3 Datový rámec pro MR-MG30.....	39
4.4 Programování a „oživení“ MR-MG30.....	42

5 Model dvouosého lineárního servopohonu.....	47
5.1 Návrh a realizace modelu.....	47
5.1.1 Parametry jednotlivých komponent modelu.....	49
5.2 Program ovládající model.....	50
5.2.1 Program pro ovládací panel PLC.....	58
Závěr.....	63
Použitá literatura.....	64
Příloha A.....	66
Příloha B.....	67
Příloha C.....	71

Seznam ilustrací

Seznam obrázků

Obrázek 1.1 - Krokový motor s proměnným magnetickým odporem.....	15
Obrázek 1.2 - Krokový motor s radiálně polarizovaným permanentním magnetem.....	16
Obrázek 1.3 - Krokový motor s axiálně polarizovaným permanentním magnetem.....	17
Obrázek 1.4 - Schéma unipolárního řízení.....	18
Obrázek 1.5 - Schéma bipolárního řízení.....	19
Obrázek 1.6 - Časové průběhy buzení krokových motorů.....	21
Obrázek 2.1 - Konstrukce lineárního motoru.....	24
Obrázek 2.2 - Princip práce dvoufázového lineárního krokového motoru.....	26
Obrázek 2.3 - Princip práce trojfázového lineárního krokového motoru.....	27
Obrázek 3.1 - Zjednodušený náčrt dvojčinného pneumatického válce.....	29
Obrázek 4.1 - Jednotlivá rozhraní řídicí jednotky MR-MG30.....	31
Obrázek 4.2 - Znak vysílaný PROFIBUSEM.....	34
Obrázek 4.3 - Druhy datových rámců.....	35
Obrázek 4.4 - Obsazení jednotlivých pinů RS-485.....	36
Obrázek 4.5 - Kabeláž a zakončovací prvek standardu RS-485.....	37
Obrázek 4.6 - Cyklická a acyklická komunikace standardu RS-485.....	38
Obrázek 4.7 - Datový rámec PPO typu 5.....	40
Obrázek 4.8 - Jednotlivá datová slova přenášeného datového rámce.....	41
Obrázek 4.9 - Obsazení jednotlivých bitů přenášeného datového rámce výstupních dat.....	41
Obrázek 4.10 - Obsazení jednotlivých bitů přenášeného datového rámce vstupních dat.....	42
Obrázek 4.11 - Inicializační sekvence MR-MG30.....	45
Obrázek 4.12 - Inicializační sekvence servoměniče.....	46
Obrázek 5.1 - Blokové schéma dvouosého lineárního servopohonu.....	48
Obrázek 5.2 - Model dvouosého servopohonu.....	49
Obrázek 5.3 - Vnitřní uspořádání modelu dvouosého servopohonu.....	49
Obrázek 5.4 - Schéma polohování jednotlivých os servopohonu.....	52

Obrázek 5.5 - Funkční blok DP SupervisorQ.....	53
Obrázek 5.6 - Funkční blok Servo1.....	55
Obrázek 5.7 - Funkční blok Servo2.....	56
Obrázek 5.8 - Funkční blok Control.....	58
Obrázek 5.9 - Ovládací panel PLC s úvodní obrazovkou.....	61
Obrázek 5.10 - Obrazovka reprezentující programový blok Polohování.....	61
Obrázek 5.11 - Obrazovka reprezentující programový blok Alarmy.....	62
Obrázek 5.12 - Obrazovka reprezentující programový blok Parametry.....	63

Seznam tabulek

Tabulka 2.1 - Použití mechanických vedení v závislosti na rychlosti.....	27
Tabulka 4.1 - Jednotlivé části řídicích dat polohování.....	31
Tabulka 4.2 - Základní specifikace MR-MG30.....	32
Tabulka 4.3 - Porovnání referenčního modelu ISO/OSI a PROFIBUS-DP.....	33
Tabulka 4.4 - Přenosová rychlost RS-485 v závislosti na délce vedení.....	35
Tabulka 4.5 - Možné zobrazené kombinace na displeji servoměniče.....	46
Tabulka 5.1 - Parametry procesoru Q02CPU.....	49
Tabulka 5.2 - Základní parametry servoměniče MR-J2S-20B.....	50
Tabulka 5.3 - Základní parametry servomotoru HC-KFS23.....	50
Tabulka 5.4 - Význam jednotlivých proměnných funkčního bloku DP SupervisorQ.....	53
Tabulka 5.5 - Význam jednotlivých proměnných funkčních bloků Servo1 a Servo2.....	56
Tabulka 5.6 - Jednotlivé proměnné funkčního bloku Control.....	58

Seznam použitých termínů a zkratek

AC	Alternating Current - střídavý proud
Array of bool	Datový typ – pole logických hodnot
Array of int	Datový typ – pole celých čísel
Bool	Boolean – datový typ – logická hodnota
CPU	Central Processing Unit – centrální procesorová jednotka
DC	Direct Current - stejnosměrný proud
DInt	Double Integer – datový typ – celé číslo (-2147483648 až 2147483647)
FBC	Function Block Diagram – diagram funkčních bloků
Flash	Paměť typu EEPROM (Electrically Erasable Programmable Read-Only Memory) – stálá paměť, jež může být elektricky vymazána nebo přeprogramována
I/O	Input/Output – vstupně výstupní zařízení
IL	Instruction List – seznam instrukcí
Int	Integer – datový typ – celé číslo (-32768 až 32767)
LD	Ladder Diagram – kontaktní schéma
LSB	Least Significant Bit – nejnižší platný bit
Master-Slave	Komunikace mezi aktivním zařízením a jemu podřízenými zařízeními
MSB	Most Significant Bit – nejvyšší platný bit
Oktet	Posloupnost osmi datových bitů nesoucích informaci v datovém rámci
PLC	Programable Logic Controller – programovatelný automat
POU	Program Organisation Unit – programový modul
Profibus	Process Field Bus – provozní sběrnice

Profibus DP	Process Field Bus Decentralized Periphery – nejrozšířenější varianta sběrnice Profibus
PWM	Pulse Width Modulation – pulsně šířková modulace
RAM	Random Access Memory – paměť s přímým přístupem
ROM	Read Only Memory – paměť pouze pro čtení
SFC	Sequential Function – sekvenční funkční diagram
SSCNET II	Servo System Controller Network – sériové komunikační rozhraní pro komunikaci se servoměniči
ST	Structure Text – strukturovaný text
Token	Pověření vysílat
Token Passing	Předávání pověření vysílat v logickém kruhu
Watch dog	Časovací jednotka („hlídací pes“)

Úvod

Vzhledem k tomu, že je v oblasti výrobních procesů kladen stále větší důraz na přesnost, musí se také odpovídajícím způsobem zdokonalovat výrobní zařízení. Proto v této oblasti hrají významnou roli i servopohony, jež musí zajistit velmi přesné polohování, ať už jsou využity v různých manipulátorech, či v jiných aplikacích. Vzhledem k tomu, že jsou používány servopohony často zastaralé nebo ne zrovna vhodně zvolené pro různé aplikace, musí se neustále hledat nové cesty, jak tyto problémy řešit a tím zdokonalovat výrobní zařízení.

Tato diplomová práce byla zadána firmou Preciosa, a.s. (Jablonec nad Nisou) za účelem prostudování vlastností, způsobů programování a aplikování řídicí jednotky MR-MG30 od firmy Mitsubishi Electric. Tato řídicí jednotka by mohla být v budoucnu využita pro nejrůznější aplikace ve výrobních procesech, kde by bylo možné využívat servosystémy, které jsou připojovány prostřednictvím rozhraní SSCNET (Servo System Controller Network), což je datová sběrnice pro komunikaci se servosystémy od firmy Mitsubishi Electric. Z tohoto důvodu byl sestaven model dvouosého lineárního servopohonu s využitím této řídicí jednotky, za účelem seznámení se s jejím programováním a následnou možností ověření získaných zkušeností na reálné modelové úloze.

Diplomová práce je rozčleněna do pěti kapitol, z nichž první tři se zabývají vlastnostmi technických prostředků pro realizaci lineárního servopohonu. Čtvrtá kapitola je zaměřena na vlastnosti a způsob programování řídicí jednotky MR-MG30 a dále na problematiku komunikačního protokolu Profibus DP. Poslední kapitola této práce se zabývá vlastním ověřením práce jednotky MR-MG30 na reálné modelové úloze.

1 Krokové motory

Krokový motor je motor s impulsním napájením a s nespojitým funkčním pohybem, jenž se děje po jednotlivých krocích. Krokové motory se ve většině případů používají pro nastavování polohy v elektrických pohonech [1].

Krokové motory rozdělujeme z pohledu řízení na krokové motory s unipolárním napájením a na krokové motory s bipolárním napájením a dále na jednofázové a vícefázové. Z pohledu konstrukce můžeme krokové motory rozdělit na motory s permanentním magnetem – motor s aktivním rotorem, a na motory s proměnným magnetickým odporem (reluktanční krokové motory) – motor s pasivním rotorem [2].

1.1 Funkce krokových motorů

Základní funkcí krokového motoru je pohyb jeho rotoru o jeden krok, kde krok chápeme jako mechanickou odezvu rotoru krokového motoru na jeden řídicí impuls. Při tomto kroku koná rotor pohyb z výchozí magnetické klidové polohy do nejbližší magnetické klidové polohy. Velikost kroku chápeme jako úhel α , jenž je přímo úměrný změně polohy rotoru po příchodu jednoho řídicího impulsu. Velikost úhlu α je závislá na konstrukci krokového motoru a na použitém způsobu jeho ovládání [1]. Pro vyvolání změny polohy napájíme jednotlivé fáze vinutí statoru v sekvencích, takže po připojení napájení na jednotlivé fáze se rotor krokového motoru snaží natočit tak, aby byl výsledný magnetický odpor minimální. Pokud je výsledný magnetický odpor minimální, znamená to, že statický úhel zátěže β je roven nule [2]. Tomuto úhlu odpovídá výchylka rotoru, jenž je nabuzen, z klidové magnetické polohy při dané zátěži na hřídeli. Pokud se tedy jedná o nezatížený krokový motor, nastaví se rotor motoru tak, že se zarovnají zuby rotoru se zuby statoru. Otáčky rotoru jsou dány kmitočtem kroků odpovídající počtu kroků vykonaných rotorem motoru za sekundu [1]. Otáčky krokového motoru určíme ze vztahu:

$$n = \frac{60 \cdot f \cdot \alpha}{360} \quad (1.1)$$

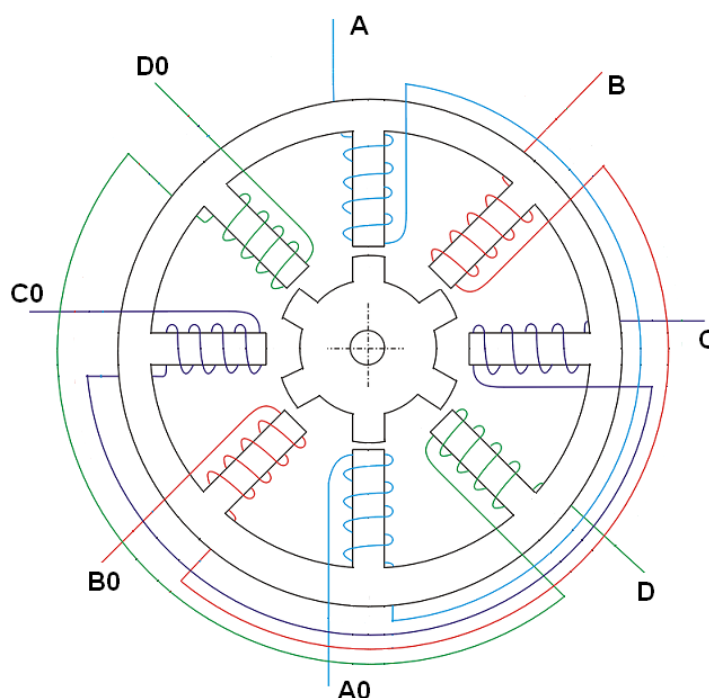
kde: n ...počet otáček za minutu
 f ...frekvence kroků [Hz]
 α ...velikost kroku [°]

1.1.1 Krokový motor s proměnným magnetickým odporem

Krokové motory s pasivním rotorem jsou konstruovány tak, že rotor krokového motoru je tvořen pouze svazkem plechů z feromagnetického materiálu, které jsou umístěny (nalisovány) na hřídel. Stator je tvořen několika póly, na nichž je vždy navinuta jedna cívka. Cívky protilehlých pólů jsou spojeny a tvoří jednotlivé fáze krokového motoru, které jsou připojeny na příslušný ovladač krokového motoru. Ovladač je tvořen komutátorem a výkonovými spínacími prvky a tento celek zajišťuje řízení chodu krokového motoru. Buzení jednotlivých fází obstarává komutátor a dále zajišťuje spínání výkonových prvků, aby se na každý řídicí impuls krokový motor natočil o jeden krok [1].

Při buzení jedné z fází krokového motoru začne cívkou protékat proud a následně se přitáhne zub rotoru (pólový nástavec), který je zrovna nejbližší ke statorovému pólu, čímž dojde k zákrytu buzených pólů statoru a zubů rotoru – v tomto stavu je minimální magnetický odpor (reluktance) [1].

Krokové motory s proměnným magnetickým odporem mají provozní točivé momenty v jednotkách mNm až v jednotkách Nm a velikosti kroku jsou obvykle velice malé: 1° až 5° [2].



Obrázek 1.1 - Krokový motor s proměnným magnetickým odporem

1.1.2 Krokový motor s permanentním magnetem

Rotor tohoto typu krokového motoru tvoří permanentní magnet, který se při změně buzení fází přizpůsobuje svým natočením tak, že se natočí, aby souhlasila polarita magnetu s magnetickým tokem statorové části krokového motoru [2]. Tyto motory mají malou časovou konstantu, z důvodu zařazeného permanentního magnetu v magnetickém obvodu. Díky tomu lze docílit vyšších provozních kmitočtů než u motorů s pasivním rotorem (jednotky až desítky kHz) [1].

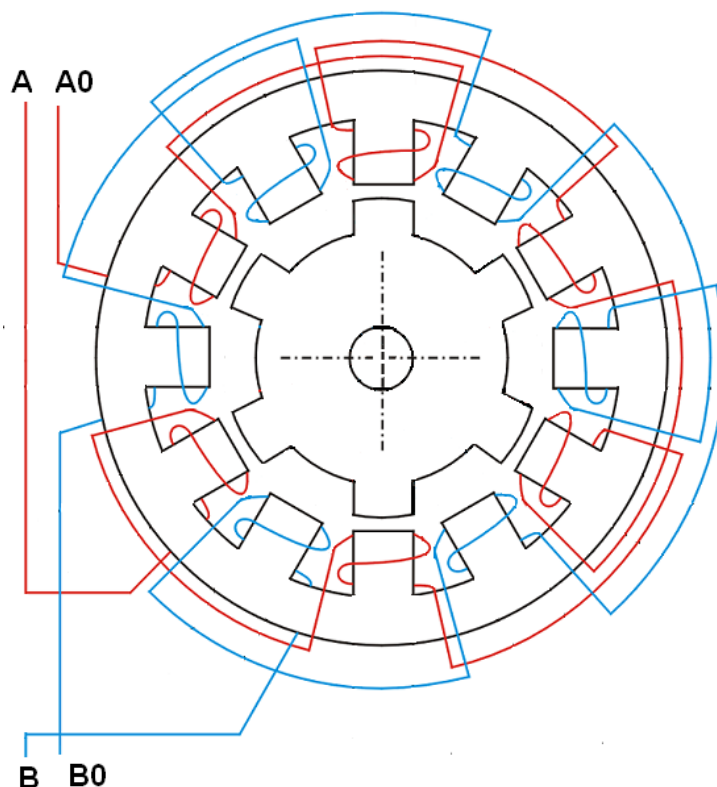
Krokové motory s permanentním magnetem můžeme rozdělit na:

- a) Krokový motor s radiálně polarizovaným permanentním magnetem
- b) Krokový motor s axiálně polarizovaným permanentním magnetem

Krokový motor s radiálně polarizovaným permanentním magnetem

Rotor krokového motoru s radiálně polarizovaným permanentním magnetem je tvořen permanentními magnety, které jsou umístěny po obvodu rotoru – střídají se

severní (S) a jižní (J) póly magnetů. Počet pólů rotoru je poloviční oproti počtu pólů statoru a navíc počet pólů statoru je dělitelný čtyřmi. Krokový motor s radiálně polarizovaným magnetem má dvoufázové vinutí, ve kterých musí být zajištěny změny směru proudu. Provozní točivé momenty se zde pohybují v řádu jednotek Nm a velikosti kroků jsou relativně velké ($> 15^\circ$) [1].



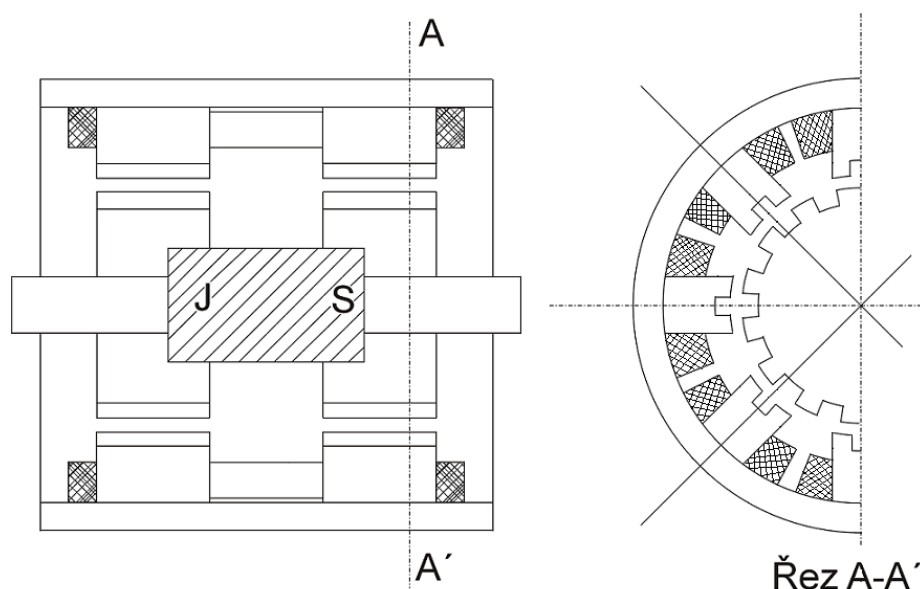
Obrázek 1.2 - Krokový motor s radiálně polarizovaným permanentním magnetem

Krokový motor s axiálně polarizovaným permanentním magnetem

Krokové motory s axiálně polarizovaným permanentním magnetem jsou někdy též označovány jako hybridní krokové motory. Rotor motoru je tvořen hřídelí z nemagnetického materiálu a dvěma pólovými nástavci složenými z plechů, které jsou na této hřídeli nalisovány. Mezi pólovými nástavci je uložen permanentní magnet, který je axiálně polarizovaný a jeho uložení je zajištěno, že každý pólový nástavec má jinou magnetickou polaritu. Pólové nástavce rotoru mají na svém obvodu vytvořeny zuby, pomocí kterých se určuje velikost kroku, a tyto pólové nástavce jsou natočeny proti

sobě o polovinu zubové rozteče v osovém směru. Buzením jednotlivých fází vinutí vzniká točivé statorové magnetické pole, které sleduje rotor krokového motoru tak, že se vždy zuby rotoru které jsou nejbližší, nastaví do magneticky klidové polohy. U této varianty krokového motoru musí být zajištěno, aby výkonové prvky umožňovali měnit směr magnetického toku v jednotlivých pólech statoru. Dále musí být možnost měnit směr budícího proudu v jednotlivých vinutích statoru a buzení musí být dvoufázové [1].

Krokové motory s axiálně polarizovaným permanentním magnetem mají relativně malý základní krok (cca od $0,36^\circ$ do 5°) a provozní točivé momenty se pohybují v řádu mNm až v desítkách Nm [2].



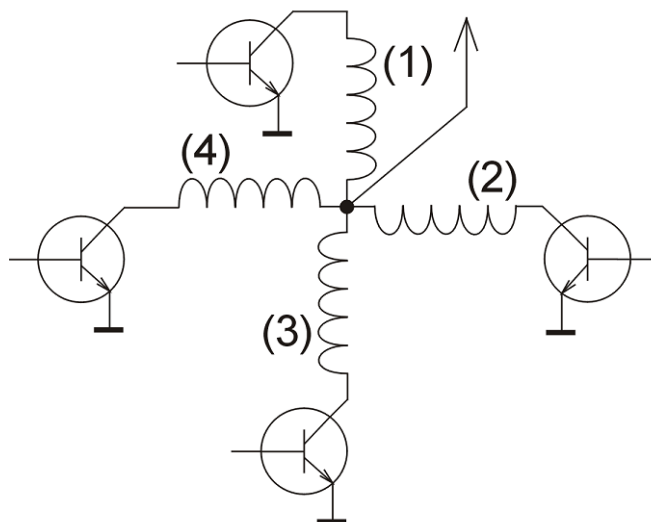
Obrázek 1.3 - Krokový motor s axiálně polarizovaným permanentním magnetem

1.2 Řízení krokových motorů

Abychom mohli řídit krokový motor, musíme zajistit napájení statorové části krokového motoru. Statorová vinutí můžeme napájet *unipolárním* nebo *bipolárním* způsobem [3].

Unipolární řízení

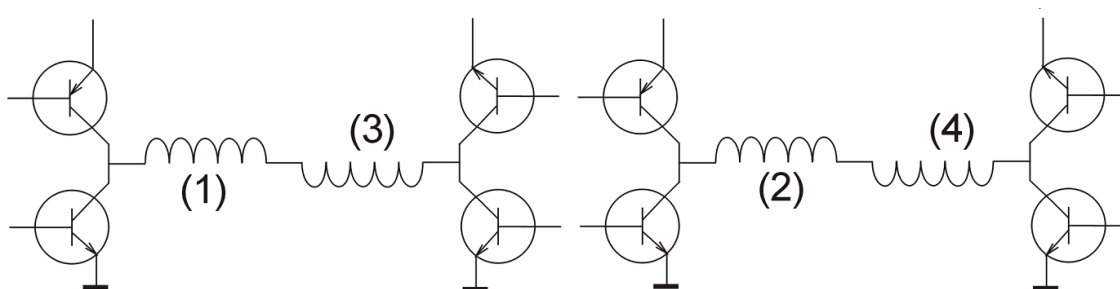
Při tomto způsobu řízení prochází proud vždy jen jednou cívkou statorového vinutí. Krokový motor řízený unipolárním způsobem má relativně malý proudový odběr, ale cenou za to je menší poskytovaný točivý moment. Ke spínání napájecího napětí jsou zapotřebí čtyři spínací prvky – tranzistory. Napájecí napětí se spíná proti zemi [3]. Zjednodušené schéma unipolárního řízení ilustruje obrázek 1.4.



Obrázek 1.4 - Schéma unipolárního řízení

Bipolární řízení

Pokud použijeme bipolární řízení krokového motoru, bude proud procházet dvěma protilehlými cívkami statorového vinutí, které jsou zapojeny s navzájem opačně orientovaným magnetickým polem. Krokový motor řízený bipolárním způsobem vykazuje větší proudovou spotřebu, avšak dosahuje větších krouticích momentů. U bipolárního typu napájení statoru je pro spínání napájecího napětí zapotřebí dvou H-můstků – pro každou větev jeden [3]. Schéma bipolárního řízení je na obrázku 1.5.



Obrázek 1.5 - Schéma bipolárního řízení

Z hlediska řízení krokových motorů rozlišujeme trojici základních principů řízení: *čtyřtaktní řízení po jedné fázi, čtyřtaktní řízení po dvou fázích a osmitaktní řízení.*

Čtyřtaktní řízení krokového motoru po jedné fázi

Pokud používáme čtyřtaktní řízení po jedné fázi, je v každé periodě řídicího signálu magnetické pole buzeno pouze jednou ze čtyř fází statorového vinutí. Při průchodu proudu vždy jednou fází vinutí statoru se rotor natočí tak, že zaujme stav minimální reluktance (magnetický odpor) magnetického obvodu. Tím se srovnají póly rotoru proti pólům statoru do souhlasné polohy, čímž se rotor dostane do magneticky klidové polohy. V této poloze je působící moment na rotor nulový. Jednotlivé fáze statorového vinutí jsou buzeny v sekvencích. Fáze, která je právě buzena (vinutím prochází el. proud) je označena číslicí 1, kdežto fáze jež buzeny nejsou, jsou označeny jako 0. Budicí sekvence jednotlivých fází jsou následující:

- 1) 1, 0, 0, 0 → el. proud prochází pouze fází A
- 2) 0, 1, 0, 0 → el. proud prochází pouze fází B
- 3) 0, 0, 1, 0 → el. proud prochází pouze fází C
- 4) 0, 0, 0, 1 → el. proud prochází pouze fází D

Cyklickým opakováním budících sekvencí je zajištěno otáčení rotoru krokového motoru [1].

Čtyřtaktní řízení krokového motoru po dvou fázích

Při čtyřtaktním řízení po dvou fázích se budí současně vždy dvě blízké fáze:

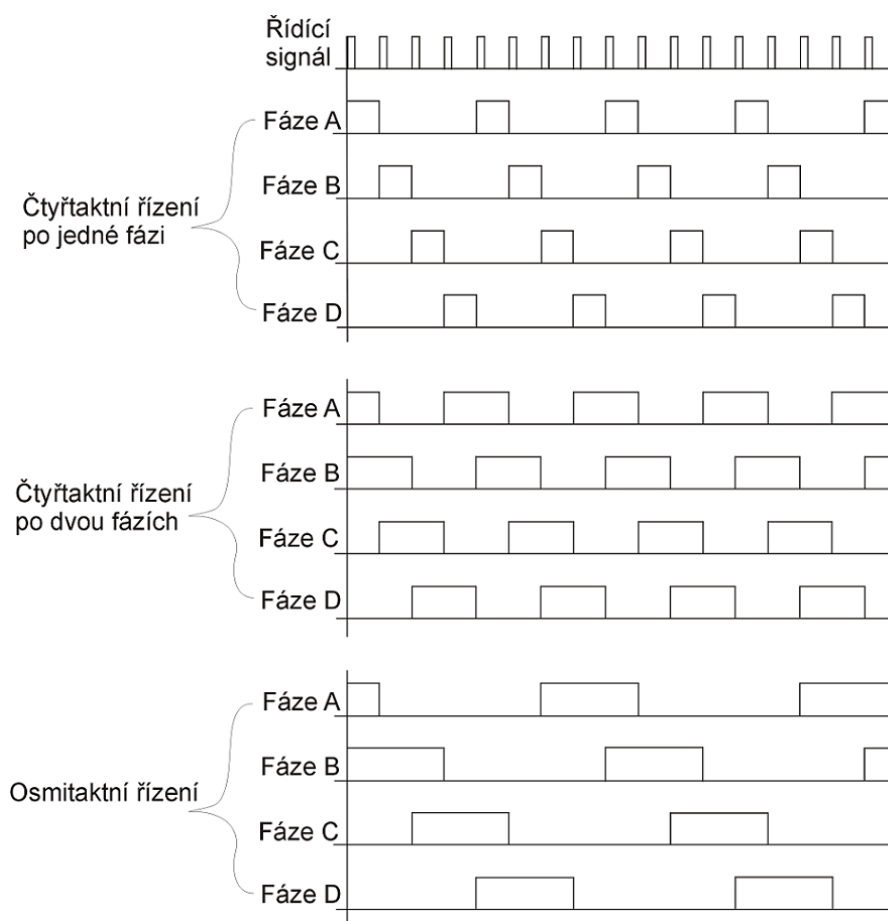
- 1) 1, 0, 0, 1 \rightarrow el. proud prochází fází A a D
- 2) 1, 1, 0, 0 \rightarrow el. proud prochází fází A a B
- 3) 0, 1, 1, 0 \rightarrow el. proud prochází fází B a C
- 4) 0, 0, 1, 1 \rightarrow el. proud prochází fází C a D

Při použití čtyřtaktního řízení po dvou fázích dostaneme stejné velikosti kroku, jako u čtyřtaktního způsobu řízení krokového motoru po jedné fázi, jelikož se pouze mění magnetická klidová poloha mezi magnetickými póly statoru. Magnetické pole statoru má vždy ustálenou polohu v geometrické ose mezi sousedními statorovými zuby [1]. Tento způsob řízení krokového motoru má však za následek dvojnásobnou proudovou a díky tomu i výkonovou náročnost, jež vede k většímu oteplování motoru, což se dá považovat za nevýhodu. Naopak výhodou tohoto způsobu řízení (oproti čtyřtaktnímu řízení krokového motoru po jedné fázi) je zvětšení statického momentu asi na dvojnásobnou hodnotu [4].

Osmitaktní řízení krokového motoru

Osmitaktní způsob řízení krokového motoru dostaneme sloučením předchozích dvou způsobů řízení, a to tak, že vložíme mezi kombinace jednofázové kombinace dvoufázové. Jelikož dochází ke střídání se zapojení jedné a dvou fází krokového motoru, mění se i v tomto rytmu velikost provozního točivého momentu motoru, proto bude celkový točivý moment oproti čtyřtaktnímu řízení po dvou fázích menší. Výhodou použití tohoto způsobu je zmenšení základního kroku motoru na poloviční hodnotu, čímž získáme větší úhlové rozlišení krokového motoru [4].

Časové průběhy jednotlivých způsobů řízení krokových motorů jsou znázorněny na obrázku 1.6.



Obrázek 1.6 - Časové průběhy buzení krokových motorů

1.3 Mikrokrokování

Pokud chceme ještě zmenšit základní kroky krokového motoru, musíme použít takzvané mikrokrokování (microsteps), kterým dosáhneme menších přírůstků (inkrementů) pohybu rotoru. Menších přírůstků dosáhneme postupným snižováním proudu v jedné fázi statorového vinutí, zatímco ve druhé fázi vinutí budeme proud postupně zvyšovat. Tím se bude výsledný vektor magnetické indukce pohybovat mezi krajními polohami, danými vybuzením jednotlivých fází. Pokud uvažujeme dvoufázový hybridní motor, je pro jeden plný krok motoru potřebná rotace magnetického pole o 90° elektrických [4]. Uvažujeme-li velikost kroku jako:

$$\Delta \Phi = \frac{2\pi}{mN} \quad (1.2)$$

kde: m ...počet fází vinutí statorů

N ...počet zubů rotoru

a dále uvažujeme-li, že se moment motoru mění sinusově s polohou rotoru a mezní vazební moment je úměrný proudu jednotlivých fází, můžeme napsat rovnice:

$$M_A = -k_T \cdot i_A \cdot \sin(N \cdot \Delta \Phi) \quad (1.3)$$

$$M_B = -k_T \cdot i_B \cdot \sin\left(N \cdot \Delta \Phi - \frac{\pi}{2}\right) \quad (1.4)$$

kde: k_T ...konstrukční konstanta krokového motoru

i_A, i_B ...proudy jednotlivých fází statorového vinutí

Pokud krokový motor pracuje s plnou délkou kroku, jsou fáze buzeny kladnou nebo zápornou hodnotou proudu o velikosti I a výsledný vektor magnetického pole bude pootočen o $\varphi = 90^\circ$ elektrických. Pokud poté uvažujeme režim mikrokrokování, můžeme tento vektor rozdělit na n mikrokroků a pro proudy jednotlivých fází pak můžeme psát rovnice:

$$i_A = I \cdot \cos\left(\frac{k\varphi}{n}\right) \quad (1.5)$$

$$i_B = I \cdot \cos\left(\frac{k\varphi}{n} - \frac{\pi}{2}\right) \quad (1.6)$$

kde: $k = 0, 1, 2, \dots$

n ...počet mikrokroků

φ ...vektor magnetického pole

i_A, i_B ...proudy jednotlivých fází statorového vinutí [4]

2 Lineární motory

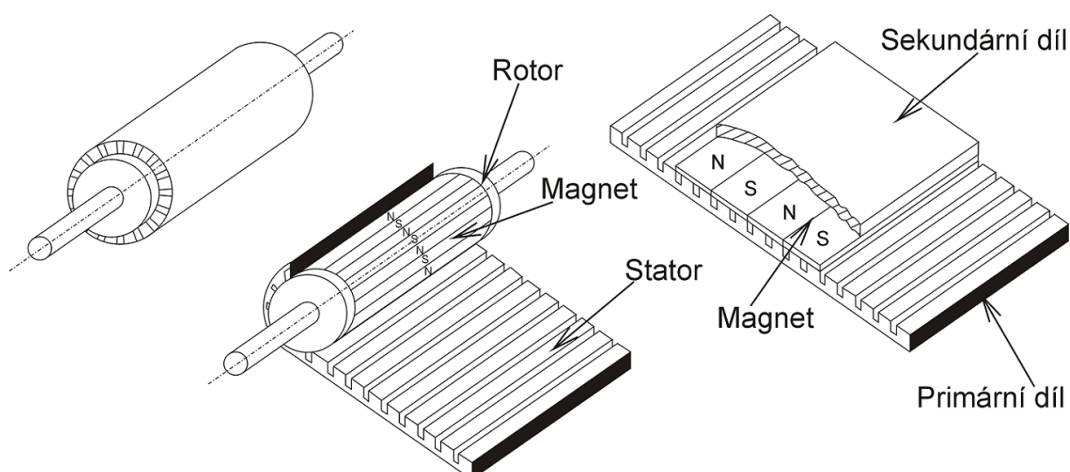
Lineární motor je motor zajišťující převod elektrické energie na lineární pohyb. Používá se zejména v těch aplikacích, kde vyžadujeme velmi přesné polohování bez překmitů.

2.1 Konstrukce lineárního motoru

Lineární motor si můžeme představit jako klasický rotační motor, ať už synchronní nebo asynchronní, který je rozvinutý do roviny (obr. 2.1), a tudíž bude konat místo rotačního pohybu pohyb lineární. Lineární motor nemá ovšem stator a rotor, jak je tomu u rotačních motorů, ale primární díl (obdobu statoru u rotačních motorů) a sekundární díl (obdobu rotoru u rotačních motorů). Primární díl u lineárních motorů tvoří svazky plechů z feromagnetického materiálu a trojfázové vinutí, které je uloženo v jeho drážkách. Sekundární část lineárního motoru zpravidla představuje delší část stroje a je tvořena permanentními magnety, které jsou nalepeny na ocelové podložce. Tyto permanentní magnety jsou většinou ze vzácných zemin (například Nd-Fe-B) [6].

U lineárního motoru může být pohyblivá jak primární část, tak i sekundární část. O tom, která z nich bude pohyblivá, rozhoduje konstrukční uspořádání poháněného zařízení, avšak ve většině případů se pohybuje primární část po libovolně dlouhé dráze tvořené libovolným počtem sekundárních částí [5].

Přivedením řídicího proudu do vinutí primární části lineárního motoru (pohyblivé části) dojde ke vzniku magnetického pole mezi primární a sekundární částí a díky tomu se začne primární část pohybovat. Rychlost pohybu primární části můžeme ovládat úrovní proudu [6].



Obrázek 2.1 - Konstrukce lineárního motoru

2.2 Synchronní lineární motor

Jde o lineární motor, který má trojfázové vinutí uložené na primární části motoru. Primární část motoru je buzena permanentními magnety, jenž jsou součástí sekundární části motoru. Permanentní magnety jsou ve většině případů ze vzácných zemin (například Nd-Fe-B), které mají špičkové magnetické parametry. Synchronní lineární motory je nutno napájet z vektorově řízených měničů kmitočtu a tudíž pro to musí být uzpůsobeny [5].

Synchronní lineární motor s primární částí bez feromagnetických materiálů

Tento typ motorů se vyznačuje nízkou hmotností. Další výhodou je absence pulsací tažné síly a prakticky nulové přitažné síly mezi primární a sekundární částí motoru. Tyto pohony se využívají zejména v aplikacích náročných na rychlost posuvu, přesnost a celkově nízkou hmotnost pohonu [5].

2.3 Asynchronní lineární motor

Asynchronní lineární motory jsou založeny na principu asynchronního stroje. Jelikož sekundární část není tvořena permanentními magnety jako v případě synchronních motorů, ale tvoří ji klec nakrátko, jde o jednodušší a levnější provedení lineárního motoru. Klec nakrátko je tvořena buď vinutím, které je uloženo do drážek sekundární části, nebo může být provedena jako hliníkový pásek připevněný na ocelové podložce. Velkou výhodou těchto motorů je, že je lze připojit přímo do standardní rozvodné sítě, bez použití jakéhokoliv měniče kmitočtu [5].

2.4 Tubulární lineární motor

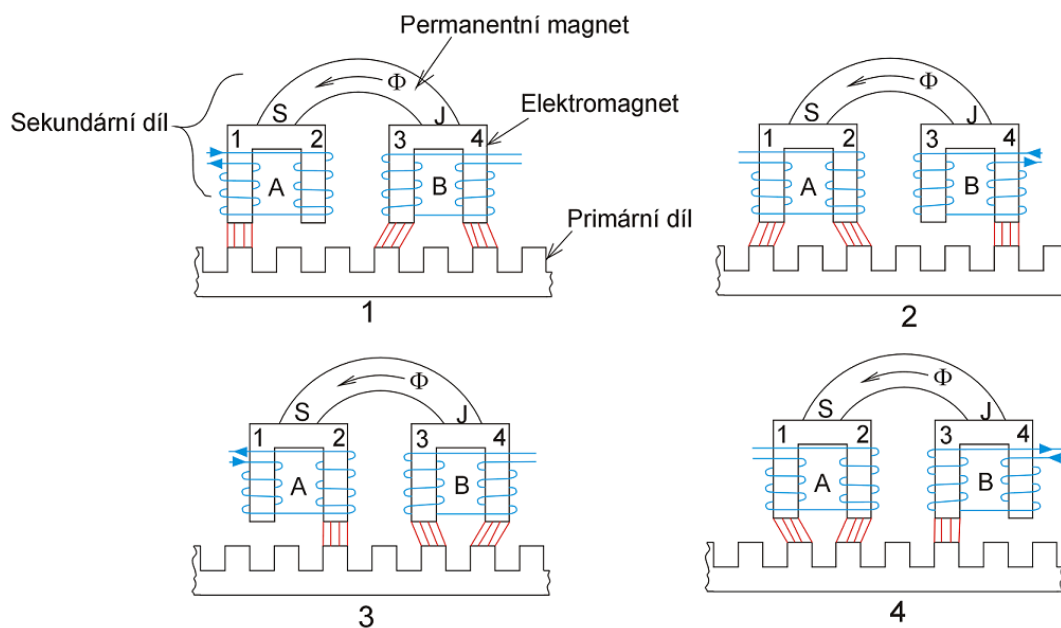
Tubulární lineární motor je složen z pevné primární části, která má tvar válce s chlazeným vinutím (většinou vodou) a sekundární části, která je pohyblivá. Sekundární část je tvořena ocelovou objímkou, na níž jsou připevněny permanentní magnety. Velkou výhodou tohoto konstrukčního uspořádání je absence pulsací tažné síly, jelikož je uspořádání motoru rotačně symetrické a tudíž dochází ke kompenzaci přitažné magnetické síly mezi primární a sekundární částí. Tím je zajištěno, že nedochází k přenosu přitažných magnetických sil na poháněné zařízení. Tyto pohony se využívají zejména v aplikacích, kde je potřebné přesné polohování – např. u přísuvů vyvrtávacích automatů desek plošných spojů [5].

2.5 Krokový lineární motor

Dvoufázový lineární krokový motor

Princip lineárního dvoufázového motoru (obr. 2.2) je následující: běžec motoru (primární část) je tvořen permanentním magnetem a dvěma elektromagnety, kde má každý dva zuby. Magnetický tok vycházející z permanentního magnetu prochází prvním elektromagnetem přes vzduchovou mezeru do statoru a odtud se opět přes vzduchovou mezeru a druhý elektromagnet vrací zpět do permanentního magnetu.

Když je cívka odpojena od zdroje proudu, bude magnetický tok procházet přes oba zuby příslušného elektromagnetu. Jakmile začne cívkou elektromagnetu procházet proud, magnetický tok bude procházet pouze jedním zubem příslušného elektromagnetu, a zuby běžce se zarovnájí se statorovými zuby. Zarovnanými zuby bude procházet maximální magnetický tok, zatímco na vedlejším zubu je zanedbatelné hodnoty [7].

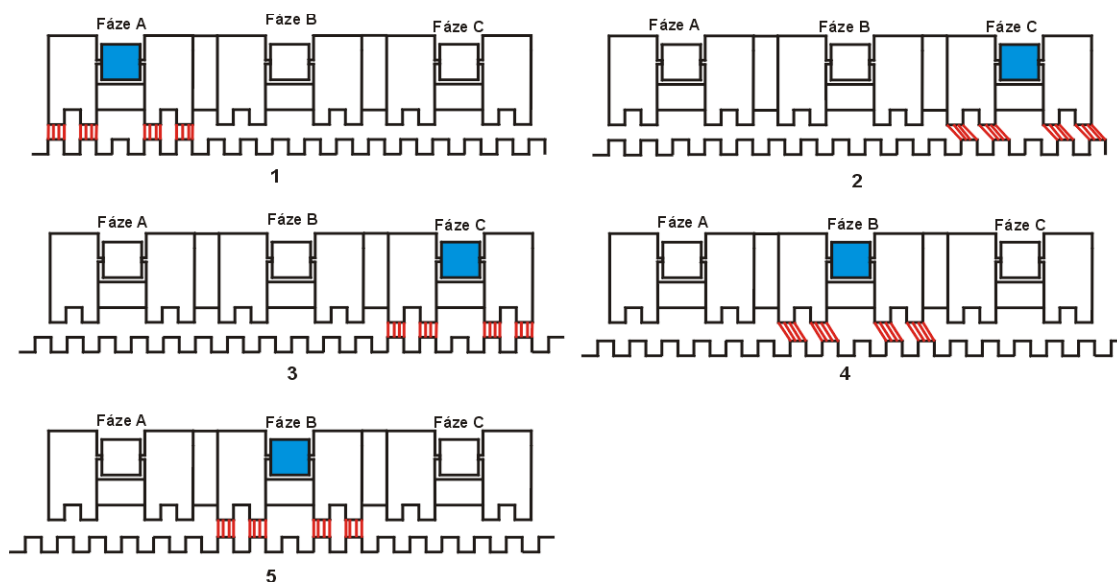


Obrázek 2.2 - Princip práce dvoufázového lineárního krokového motoru

Trojfázový lineární krokový motor

Princip trojfázového lineárního krokového motoru je velice podobný principu dvoufázového lineárního krokového motoru (Obr. 2.2). Nejprve dojde k buzení první fáze a díky tomu dojde k zarovnání zubů statoru se zuby běžce – tzv. inicializační krok. V následujícím kroku vyšle řídicí jednotka řídicí impuls do jednoho z dalších dvou elektromagnetů, v závislosti na požadovaném směru pohybu běžce. Pokud budeme uvažovat požadovaný pohyb běžce doprava, vyšle se řídicí impuls do třetího elektromagnetu, čímž dojde k buzení cívky. Tím začne procházet magnetický tok permanentního magnetu a třetího elektromagnetu přes vzduchovou mezeru do zubů statoru a také působením magnetických sil dojde k zarovnání zubů. Procházející magnetický tok bude mít minimální odpor (odpor tvoří jen vzduchová mezera). Pro

další krok a pohyb běžce doprava bude vybuzena cívka prostředního elektromagnetu a stejným principem dojde opět k zarovnání zubů statoru a elektromagnetu, čímž běžec vykoná další krok.[7].



Obrázek 2.3 - Princip práce trojfázového lineárního krokového motoru

2.6 „Bezželezný“ lineární motor

Konstrukce „bezželezného“ lineárního motoru se vyznačuje tím, že neobsahuje žádné magneticky aktivní železo v primární části a že jeho cívky jsou umístěny v nosiči z nemagnetického materiálu. Tyto motory jsou vhodné pro aplikace, kde jsou zátěže nižších hmotností, ale současně je žádána vysoká dynamika. „Bezželezné“ lineární motory jsou schopny dosahovat vysokých hodnot rychlostí a zrychlení a umožňují velmi přesné polohování bez překmitů. Nevýhodou „bezželezných“ lineárních motorů je menší trvalé zatížení motoru, z důvodu zhoršeného odvodu vznikajícího tepla. Naopak výhodou je velmi hladký průběh pohybu i při malých rychlostech (motory „se železem“ mají pohyb při malých rychlostech trhaný) [8].

2.7 Příslušenství lineárních motorů

Lineární pohon se skládá z vlastního motoru a z napájecího zdroje. K těmto dvěma základním částem se ještě připojují mechanické vedení, přídatné chladiče a řídicí systém, jenž se skládá nejméně z regulátoru a snímače polohy [5].

Mechanické vedení

Použití jednotlivých mechanických vedení v závislosti na rychlosti uvádí následující tabulka:

Použité mechanické vedení	Maximální rychlost [ms^{-1}]
Kluzné kovové plochy	0,5
Kuličková pouzdra	1,0
Lineární kuličková nebo válečková ložiska	10,0
Keramické kluzné plochy	20,0
Vzduchová ložiska	100,0

Tabulka 2.1 - Použití mechanických vedení v závislosti na rychlosti

Odměřovací systémy

Odměřovací systémy se nejčastěji používají inkrementální, které pracují buďto na reluktančním, magnetickém nebo fotoelektrickém principu. Reluktanční a magnetický snímač je složen ze snímací magnetické hlavičky a nosného pásku, jež má záznamovou vrstvu, která nese informaci o poloze – jedná se vlastně o magnetickou mřížku. Optické snímače jsou složeny ze snímací hlavičky (světelný zdroj a optická soustava) a kovového pásku, jenž má na svém povrchu vypálený systém rysek s roztečí 20 μm nebo 40 μm [5].

Řídicí systém – regulátory

Pro regulaci lineárních motorů se používají většinou číslicové regulátory, které mají uspořádání do tří zpětných vazeb – proudová, rychlostní a polohová. Dále se ještě v některých případech přidává regulace zrychlení [5].

3 Pneumatický servopohon

Pneumatický servopohon je často používán v nejrůznějších průmyslových aplikacích, nejčastěji v manipulačních zařízeních. V dnešní době je možné spojitě řídit rychlost pohybu pístu pneumatického válce a díky tomu je možné vytvořit pneumatický servopohon [9].

Dvěma základními stavebními kameny pro sestavení pneumatického servopohonu jsou pneumatický válec a dále pneumatický servoventil, pomocí kterého můžeme spojitě řídit tok (hmotnostní) vzduchu do pracovních prostor pneumatického válce [9].

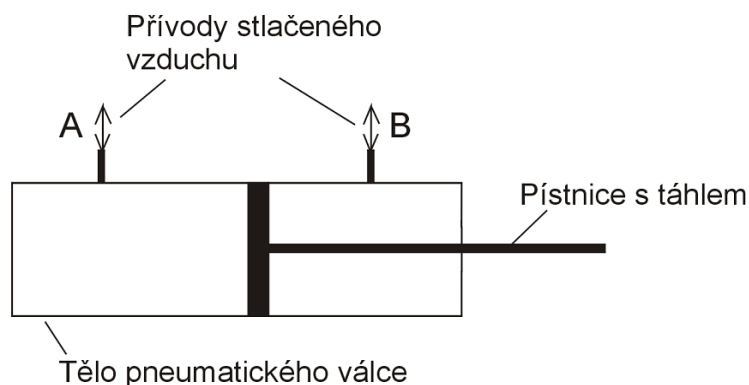
3.1 Pneumatický válec

Pneumatický válec je další možností, jak zrealizovat pohon s lineárním pohybem. Pneumatický válec využijeme zejména tam, kde není třeba často měnit parametry a nastavení stroje, a kde vyžadujeme maximální sílu v malém montážním prostoru [10].

Pokud si pneumatický válec představíme zjednodušeně, pak se skládá ze tří základních částí: 1) tělo pneumatického válce – většinou vyrobeno z hliníku

2) pístnice válce s táhlem

3) přípojky pro přívod stlačeného vzduchu do válce



Obrázek 3.1 - Zjednodušený náčrtek dvojčinného pneumatického válce

Táhlo válce s pístnicí je pohyblivé a je utěsněné vzhledem ke stěnám těla válce. Pohyb táhla je ovládán pomocí dvou přípojek pro stlačený vzduch, které jsou označeny jako A a B. V našem případě se tedy jedná o dvojčinný pneumatický válec – píst se pohybuje díky zaplňování komor válce stlačeným vzduchem – směr pohybu pístu válce určuje přípojka pro stlačený vzduch, která je právě aktivní. Dalším typem pneumatického válce je jednočinný válec. V tomto případě se píst válce pohybuje pomocí stlačeného vzduchu pouze jedním směrem. Ke zpětnému chodu pístnice je nejčastěji použita pružina [11].

Pohyb táhla s pístnicí pneumatického válce můžeme popsat diferenciální rovnicí druhého řádu – pohybovou rovnicí:

$$m \ddot{x} = F_A - F_B - (F_z + F_T) \quad (3.1)$$

$$m \ddot{x} = S_A p_A - S_B p_B - F \quad (3.2)$$

kde: x ... poloha pístu pneumatického válce

F ... výsledná síla, jež zahrnuje zatěžující sílu a třecí sílu

p_A, p_B ... tlaky v pracovních prostorech pneumatického válce

S_A, S_B ... plošné průřezy pracovních oblastí pneumatického válce

3.2 Pneumatický servoventil

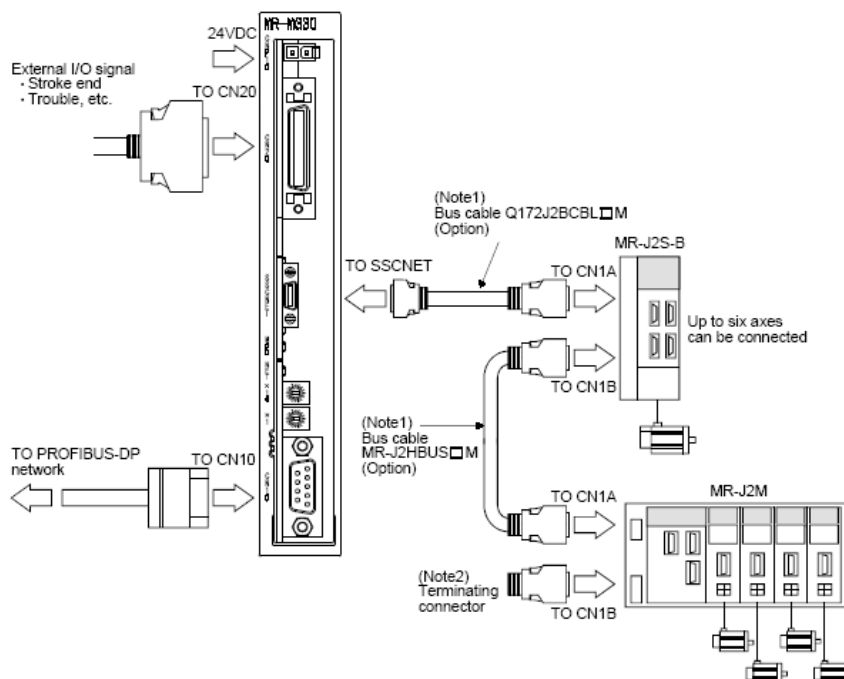
Pneumatický servoventil je nedílnou součástí pneumatického servopohonu. Ten by měl umožnit spojitě řízení průtoku vzduchu do pracovních oblastí pneumatického válce [9]. Pomocí servoventilu můžeme například spojitě řídit rychlost pneumatického válce a tím i jeho polohu. Hlavní částí ventilu je keramické šoupátko, které je drženo ve střední poloze permanentním magnetem. K vychylování šoupátka z této střední polohy dochází pomocí elektromagnetu a aktuální poloha šoupátka je bezkontaktně snímána. Součástí ventilu je i zpětnovazební elektronický regulační obvod, který porovnává skutečnou polohu šoupátka s žádanou hodnotou polohy (žádaná hodnota je ve formě napětí, či proudu) a na základě těchto dvou údajů polohu koriguje [12].

4 Řídicí jednotka MR-MG30

Jedná se o řídicí jednotku od firmy Mitsubishi Electric, která byla navržena pro komunikaci se servoměniči a k jejich řízení. Jednotka byla navržena pro připojení až šesti servoměničů, kompatibilních s touto jednotkou – možnost řízení až šestiosého servopohonu.

4.1 Funkce a parametry MR-MG30

Jednotka MR-MG30 komunikuje s PLC pomocí sběrnice PROFIBUS-DP. K sběrnici PROFIBUS-DP mohou být připojována další jednotlivá zařízení prostřednictvím standardu RS-485. Ke komunikaci mezi jednotkou a jednotlivými servoměniči dochází pomocí rozhraní SSCNET II (Servo System Controller Network). Rozhraní SSCNET bylo navrženo firmou Mitsubishi Electric speciálně pro tyto čely a můžeme si ho představit jako jednoduchou síť (sériová komunikace). Jednotku MR-MG30 a její rozhraní názorně ilustruje obrázek 4.1 – více v [13].



Obrázek 4.1 - Jednotlivá rozhraní řídicí jednotky MR-MG30 [13]

Jednotka MR-MG30 může pracovat s daty polohy buďto v absolutním, nebo přírůstkovém tvaru. Řídicí data polohy se skládají z částí, které popisuje tabulka 4.1.

Jméno	Rozsah nastavení	Jednotky
Data polohy	-999999 až +999999	x 0,001 [mm] x 0,01 [mm] x 0,1 [mm] x 1 [mm]
Rychlost servomotoru	0 až max. rychlost motoru	[ot/min]
Časová konstanta zrychlení	0 – 20000	[ms]
Časová konstanta zpomalení	0 – 20000	[ms]
Pomocná funkce	0 – 1	0: Absolutní tvar dat polohy 1: Přírůstkový tvar dat polohy

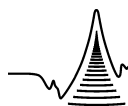
Tabulka 4.1 - Jednotlivé části řídicích dat polohování

Pokud jednotka komunikuje se servoměniči, které jsou kompatibilní s rozhraním SSCNET II, je možné obě časové konstanty z tabulky 4.1 (zrychlení a zpomalení) měnit v reálném čase. Servoměniče, které SSCNET II podporují a byly navrženy pro práci s využitím řídicí jednotky MR-MG30, jsou rovněž od firmy Mitsubishi Electric a lze je vybírat ze dvou následujících sérií:

- MR-J2 Super series
- MR-J2M series

Každá série obsahuje několik modelů servoměničů (např. MR-J2S-100B), které se liší hlavně v parametru maximálního výkonu, který nás omezuje v připojení jednotlivých servomotorů (např. do 1 kW) [13].

Jednotlivé parametry a základní specifikaci jednotky MR-MG30 popisuje tab. 4.2.



Položka		Popis
Model		MR-MPB06
Napájecí Zdroj	Napětí	24 VDC
	Povolené kolísání napětí	24 VDC \pm 10%
	Kapacita napájecího zdroje	10 W
Rozhraní napájení		24 VDC \pm 10%, 500 mA nebo více
Rozhraní	Příkaz (řízení)	PROFIBUS-DP V0
	Servo měnič	SSCNET II
Provozní teplota		0 - +55 °C
Vibrace okolního prostředí		5,9 [m/s] nebo méně
Hmotnost		0,5 [kg]

Tabulka 4.2 - Základní specifikace MR-MG30

Pokud máme k jednotce MR-MG30 připojen více než jeden servoměnič, musíme počítat s omezením řízení jednotlivých servoměničů, protože jednotka neumožňuje řídit současně více os najednou. Tudiž pokud chceme začít pracovat s nějakou z dalších možných připojených os, musíme počkat, než právě aktivní osa dokončí svou práci – přijme data od jednotky MR-MG30 a vrátí zpět zprávu o provedení.

4.2 PROFIBUS-DP

PROFIBUS DP (Decentralized Periphery) je v praxi nejběžněji používaná varianta průmyslové sběrnice. Nejčastěji je používána pro komunikaci mezi řídicími jednotkami a jednotlivými periferiemi. Řídicími jednotkami jsou nejčastěji PLC nebo průmyslová PC a pod jednotlivými periferiemi si můžeme představit různá I/O zařízení, snímače, převodníky fyzikálních veličin, atd [14].

PROFIBUS DP vychází z referenčního modelu ISO/OSI. Tento model definuje komunikaci mezi jednotlivými stanicemi (periferiemi) a řídicím systémem (řídicí jednotkou). Dále definuje jednotlivé elementy, struktury a úlohy, jež jsou nezbytné pro komunikaci. V neposlední řadě model ISO/OSI definuje jednotlivé vrstvy, které jsou hierarchicky uspořádány – viz tab. 4.3.

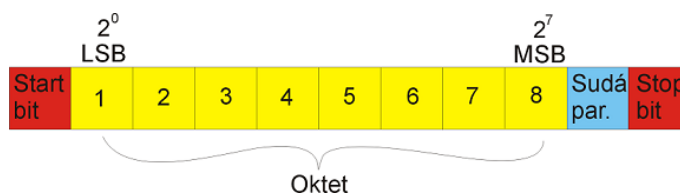
Vrstva	ISO/OSI Model	PROFIBUS DP	Popis
7	Aplikační (Application)	Aplikační (Application)	Aplikační protokol
6	Prezentační (Presentation)	Prázdné	Interpretace (kódování) dat pro následující vrstvu
5	Relační (Session)	Prázdné	Zavádí a ukončuje dočasná spojení stanic; Synchronizace komunikačního procesu
4	Transportní (Transport)	Prázdné	Kontrola vysílaných dat pro vrstvu 5 (Chyby přenosu)
3	Síťová (Network)	Prázdné	Zavádí a ukončuje spojení; Hlídá, aby nedošlo k zahlcení sítě
2	Linková (Data Link)	Linková (Data Link)	Přenosový protokol (Medium Acces Control)
1	Fyzická (Physical)	Fyzická (Physical)	Definuje přenosové médium, způsob kódování dat a přenosovou rychlost

Tabulka 4.3 - Porovnání referenčního modelu ISO/OSI a PROFIBUS-DP

Nejnižše postavenou vrstvou je vrstva fyzická (vrstva 1), nad níž jsou nadřazeny vrstvy s vyšším postavením, z nichž nejvyšší postavení má vrstva aplikační (vrstva 7). V této hierarchii podřízená vrstva poskytuje služby vrstvě vyšší – např. fyzická vrstva poskytuje služby vrstvě linkové, která poskytuje služby vrstvě síťové. PROFIBUS-DP využívá na rozdíl od referenčního modelu ISO/OSI pouze tři vrstvy – fyzickou (vrstva 1), linkovou (vrstva 2) a aplikační vrstvu (vrstva 7) [14] – viz tab. 4.3.

Aplikační vrstva zajišťuje služby, které jsou potřebné pro komunikaci mezi jednotlivými zařízeními [14].

Linková vrstva řídí přístup na sběrnici (Medium Access Control) a dále dekóduje příchozí rámce, nebo sestavuje vysílané rámce dat. Rámcem dat rozumíme posloupnost znaků, jež vysílá PROFIBUS. Vysílaný znak je jedenáctibitový a obsahuje jeden start bit, osm datových bitů → oktet, jeden sudý paritní bit a jeden stop bit [14] – viz obr. 4.2.



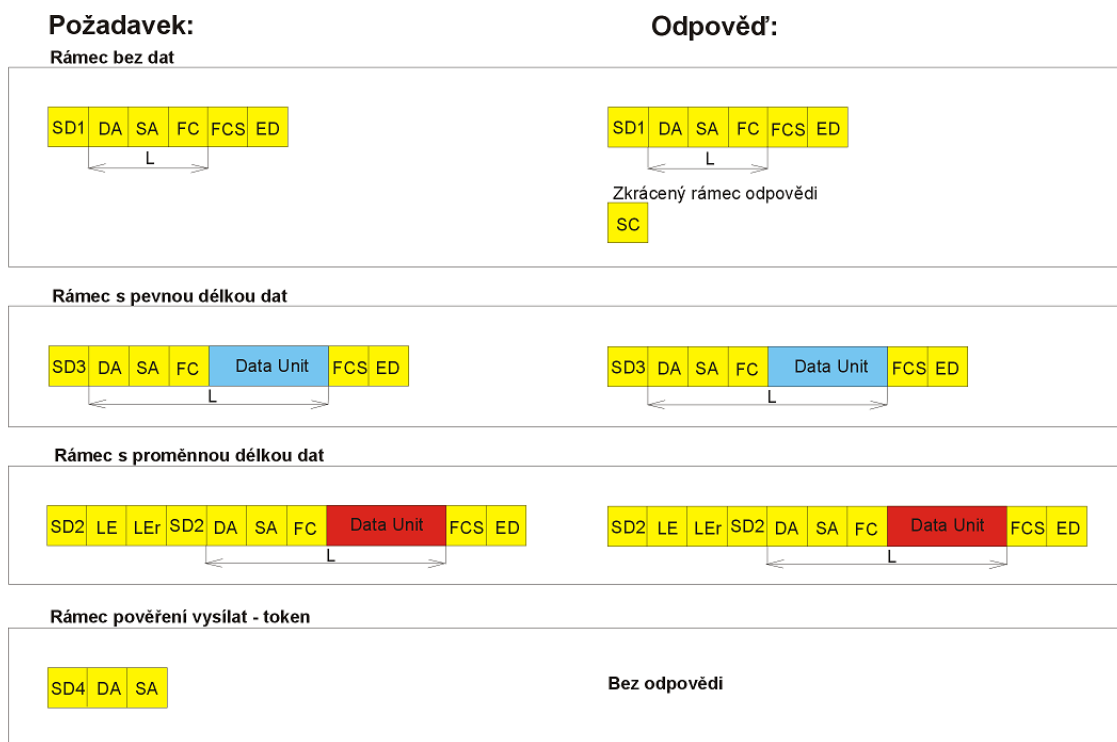
Obrázek 4.2 - Znak vysílaný PROFIBUSEM

Mezi jednotlivými oktety sestavovanými do rámců nesmějí být žádné mezery (časové).

Druhy datových rámců:

- 1) Rámec pověření vysílat – token
- 2) Rámec bez dat
- 3) Rámec s pevnou délkou dat
- 4) Rámec s proměnnou délkou dat

Každý vysílaný datový rámec je vyslán jako požadavek a po jeho vyslání by měla přijít příslušná odpověď. Každému rámcí požadavku předchází ještě synchronizační perioda - dlouhá minimálně 33 bitů [15]. Významy jednotlivých znaků použitých v datových rámcích jsou vysvětleny v příloze A.



Obrázek 4.3 - Druhy datových rámců

Fyzická vrstva definuje vlastnosti přenosového kanálu a fyzické spojení jednotlivých připojených zařízení. Pro realizaci fyzické vrstvy se nejčastěji používá přenosové technologie RS-485 (i v našem případě), která využívá stíněnou kroucenou dvojlinku. RS-485 má přenosovou rychlost od 9,6 kbit/s do 12 Mbit/s. Přenosová rychlost je přímo závislá na použité délce vedení, a na parametrech vodiče, použitého pro realizaci

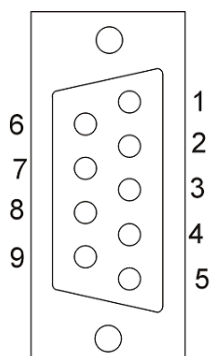
sběrnice – viz tab. 4.4.

Přenosová rychlost [Kbit/s]	Délka vedení [m]	Uvedené hodnoty odpovídají vodiči typu A, jenž má následující parametry:	
9,6; 19,2; 45,45; 93,75	1200	Impedance	135 až 165 Ω
187,5	1000	Kapacita	≤ 30 pF/m
500	400	Odpor smyčky	≤ 110 k Ω /km
1500	200	Průměr vodiče	$> 0,64$ mm
3000; 6000; 12000	100	Příčný průřez jádra	$> 0,34$ mm ²

Tabulka 4.4 - Přenosová rychlost RS-485 v závislosti na délce vedení

Tato realizace sběrnice umožňuje přidání, nebo odebrání jednotlivých stanic z (do) systému, bez ovlivnění ostatních stanic. Dalšími možnostmi realizace fyzické vrstvy jsou RS-485-IS, nebo optickými vlákny. V případě RS-485-IS se jedná o novější specifikaci RS-485. Tato varianta využívá pro přenos kabel skládající se ze čtyř vodičů v ochranné izolaci typu EEx-i, pro použití v prostředích s nebezpečím výbuchu. Optická vlákna se používají zejména pokud potřebujeme vytvořit komunikační spojení na delší vzdálenosti a v prostředích s vysokým elektromagnetickým rušením [14].

Při použití sběrnice PROFIBUS DP lze připojit až 127 stanic (zařízení) na sběrnici, avšak jednotlivé stanice musí být umístěny v tzv. segmentech, kde každý segment může obsahovat maximálně 32 stanic. Jednotlivé stanice jsou připojovány prostřednictvím klasického devítipinového D-Sub konektoru [13] – viz obr. 4.4.

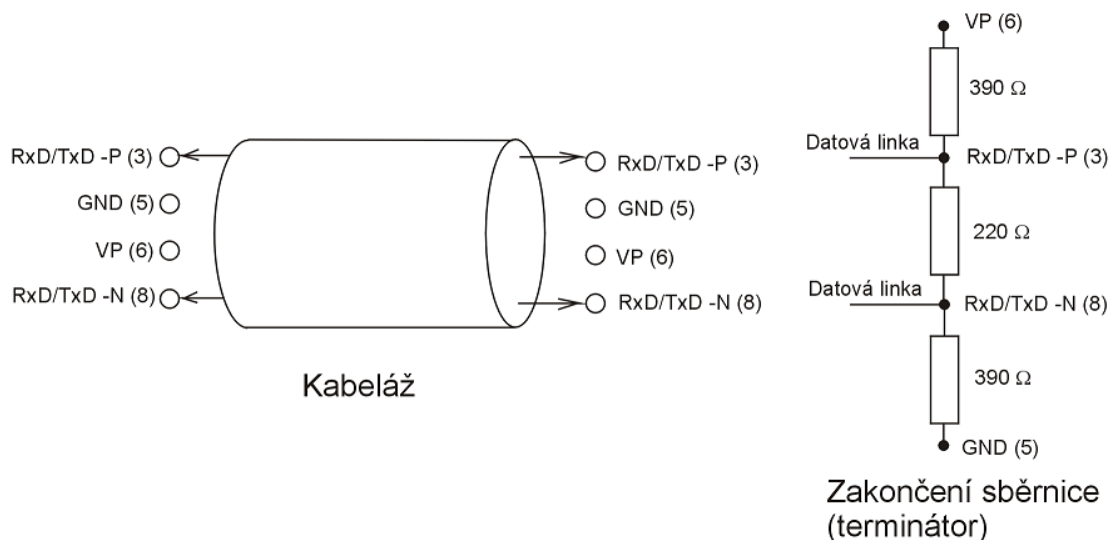


Pin	Signál	Popis
1	Stínění	Stínění
2	N/C	Nepřipojeno
3	RxD/TxD - P	Kladný signál pro vysílání a příjem dat
4	RTS	Řídicí signál - pro opakovače
5	DGND	Datová zem (Data ground)
6	VP	Napájení +5V
7	N/C	Nepřipojeno
8	RxD/TxD - N	Záporný signál pro vysílání a příjem dat
9	N/C	Nepřipojeno

Obrázek 4.4 - Obsazení jednotlivých pinů RS-485

Pokud připojíme více než 32 stanic, máme tedy více než jeden segment, musíme mezi

jednotlivé segmenty také připojit tzv. opakovač (repeater), který nám jednotlivé segmenty propojí a dále zajišťuje regeneraci signálu a galvanické oddělení jednotlivých segmentů. Další nezbytnou součástí sběrnice jsou tzv. terminátory (zakončovací prvky), kterými musí být každý začátek a konec segmentu na sběrnici zakončen – viz obrázek 4.5. Oba z dvojice použitých terminátorů pro jeden segment mají permanentní napájecí napětí +5 V [15].



Obrázek 4.5 - Kabeláž a zakončovací prvek standardu RS-485

Připojené jednotky (zařízení) ke sběrnici rozdělujeme do následujících skupin:

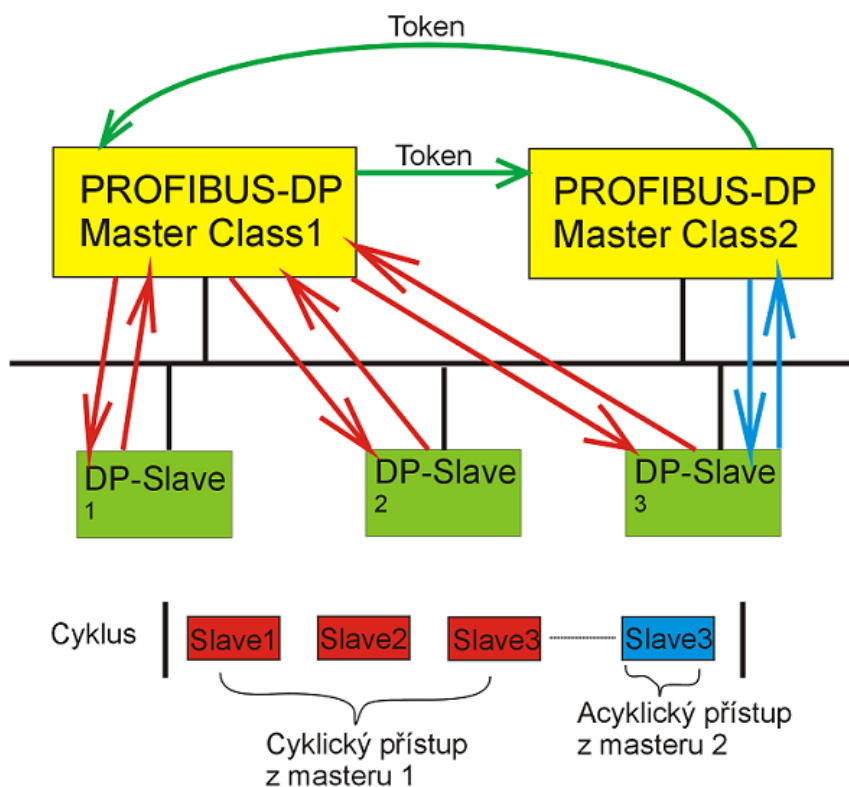
- DP Master Class1 (DPM1) – realizuje komunikaci s připojenými jemu podřízenými jednotkami (zařízeními) typu DP Slave.
- DP Master Class2 (DPM2) – realizuje diagnostické funkce a monitoring na sběrnici při uvádění sběrnice do provozu, nebo pokud je sběrnice v režimu pro účely monitoringu.
- DP Slave – jednotlivá periferní zařízení - podřízené jednotky jednotce master; slouží k získávání vstupních informací pro jednotku master, nebo k poskytování výstupních informací, které jsou posílány od jednotky master.

Ke každé sběrnici musí být připojena nejméně jedna řídící jednotka – master (DPM1).

Jednotka master má za úkol řídit provoz na sběrnici a realizovat komunikaci s jednotlivými připojenými zařízeními, ať už s podřízenými jednotkami slave, nebo s jinou jednotkou master (DPM1 nebo DPM2). Pokud jednotka typu slave není jednotkou

master přímo vyzvána aby vysílala, může data pouze přijímat. Ke komunikaci jednotky master s jednotkami slave dochází cyklicky, zatímco pokud dochází k monitoringu nebo diagnostice, jedná se o acyklické služby [14].

Pokud je na sběrnici připojena pouze jedna jednotka typu master a několik jednotek typu slave, dochází pouze ke komunikaci typu master-slave. Pokud ale bude na sběrnici více jednotek typu master, ať už několik DPM1 nebo jeden DPM1 a několik DPM2, musí docházet ke komunikaci master-slave, ale i ke komunikaci typu token passing. Připojené jednotky typu master si mezi sebou musí předávat pověření vysílat-tzv. token. Token je vždy předáván od jednotky master s nižší adresou jednotce master s vyšší adresou. Tím nám vzniká logický kruh – jednotka master s nejvyšší adresou předává token jednotce master s nejnižší adresou [15] – viz obr. 4.6.



Obrázek 4.6 - Cyklická a acyklická komunikace standardu RS-485

Každá jednotka master typu DP Master Class 1 (DPM1) musí sledovat dění na sběrnici a aktualizovat, od které jednotky typu master bude token přebírat a které ho následně pošle dál. Při konfiguraci sběrnice je definován časový interval T_{TR} , jenž udává

dobu vymezenou pro oběh pověření vysílat (token). Dalším časovým intervalem je T_{RR} , který si měří a uchovává každá jednotka master sama. Časový interval T_{RR} udává dobu, jež uplynula od doby, kdy jednotka master naposled předala pověření vysílat jednotce master s vyšší adresou. Na základě těchto dvou časových intervalů (T_{TR} a T_{RR}) si pak jednotka master vypočte čas, přidělený pro vysílání požadavků na jednotky typu slave, případně pro komunikaci s jinou jednotkou master tím, že provede porovnání těchto dvou intervalů ($T_{TR} - T_{RR}$) a výsledný časový interval je pak označován jako T_{TH} . Po tomto časovém intervalu může jednotka master token podržet. Po jeho uplynutí, musí předat token dál. Pokud by se stalo, že jednotka master hned poté co přijme token zjistí, že už nemá žádný čas pro komunikaci s ostatními jednotkami – tzn. že interval T_{RR} překročil dobu T_{TR} , může jednotka master pouze vykonat jednu komunikaci s jinou jednotkou, která má vysokou prioritu. Z tohoto důvodu musí jednotka master před každou komunikací s ostatními jednotkami zjišťovat, zda má ještě potřebný čas pro vysílání. Komunikace jednotky master s jednotkami typu slave je rozdělena do třech fází:

- Parametrizace
- Konfigurace
- Doprava dat [15]

Veškerá komunikace na sběrnici probíhá dle komunikačního protokolu příslušné verze. Verze protokolu DP jsou tři – DP-V0, DP-V1 a DP-V2. Vzhledem k tomu, že v našem případě je použita verze DP-V0, budu zmiňovat pouze její vlastnosti. Mezi základní vlastnosti DP-V0 patří cyklické čtení vstupních informací od jednotky typu slave a zapisování cyklických výstupních informací taktéž do jednotky slave. Časový cyklus sběrnice musí být kratší, než časový cyklus centrálního automatického systému. Ve většině případů je tento časový cyklus přibližně 10ms. Dalšími funkcemi protokolu DP-V0 jsou již dříve zmiňované diagnostické funkce.

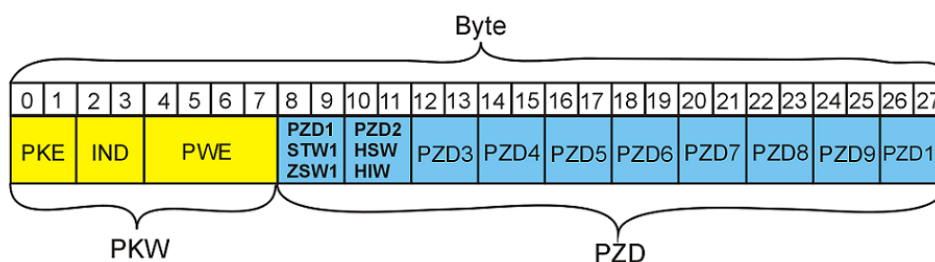
Diagnostické funkce:

- Diagnostika zařízení (stanic) – zprávy o běžných operačních stavech stanic – např. nízké napětí
- Diagnostika modulů – zprávy o nevyřešených diagnostikách nějakého I/O zařízení
- Diagnostika jednotlivých kanálů – zprávy o chybách spojovaných přímo s

nějakými I/O bity (kanály) - např. zkrat na výstupu [15]

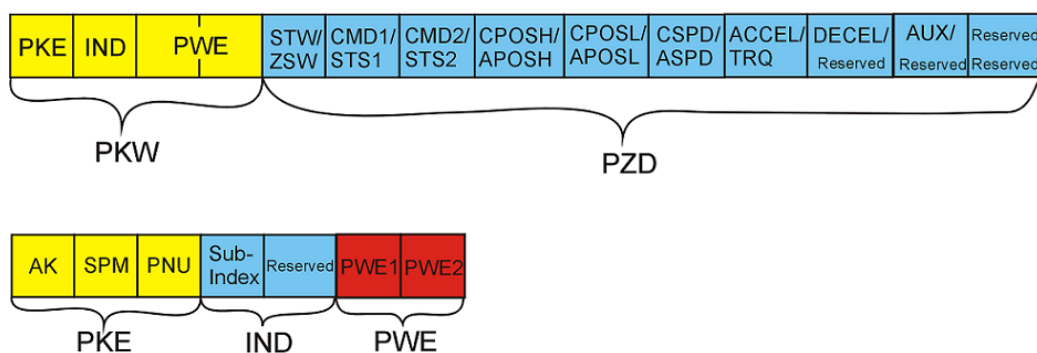
4.3 Datový rámec pro MR-MG30

Jednotka MR-MG30 v rámci komunikace s jednotlivými připojenými zařízeními vysílá datový rámec, jenž obsahuje 14 slov pro vstupní data a 14 slov pro výstupní data. Jednotlivá slova datového rámce jsou 16-ti bitová. Výstupní data jsou chápána jako požadavek o příkaz, nebo úlohu, kdy jednotka vyšle příslušnému zařízení co vše a za jakých podmínek má vykonat. Vstupní data můžeme chápat jako odezvu, kdy jednotka obdrží informace, zda bylo vše v pořádku provedeno, nebo zda došlo k nějaké chybě. Dále jsou vstupní data jednotkou využívány například pro zjištění aktuální rychlosti jednotlivých servomotorů, okamžitého točivého momentu a podobně. Datový rámec jako celek je kompatibilní s datovým rámcem PPO (Parameter Process data Object) typu 5 [16], jenž je zobrazen na obrázku 4.7.



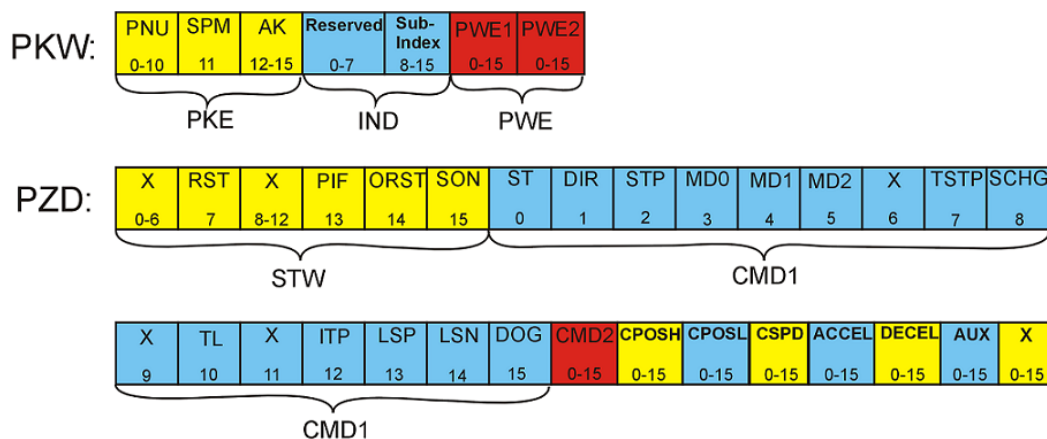
Obrázek 4.7 - Datový rámec PPO typu 5

První 4 slova datového rámce jsou sdruženy jako PKW, kde se jedná o charakteristické hodnoty předávaných parametrů. Tyto slova jednotlivé parametry nastavují a dále slouží i k případné diagnostice. Zbývajících 10 slov je sdružováno jako PZD a v jejich případě se jedná o data daného procesu [16]. Jednotlivá slova v datovém rámci, jenž posílá jednotka MR-MG30 ilustruje obrázek 4.8.

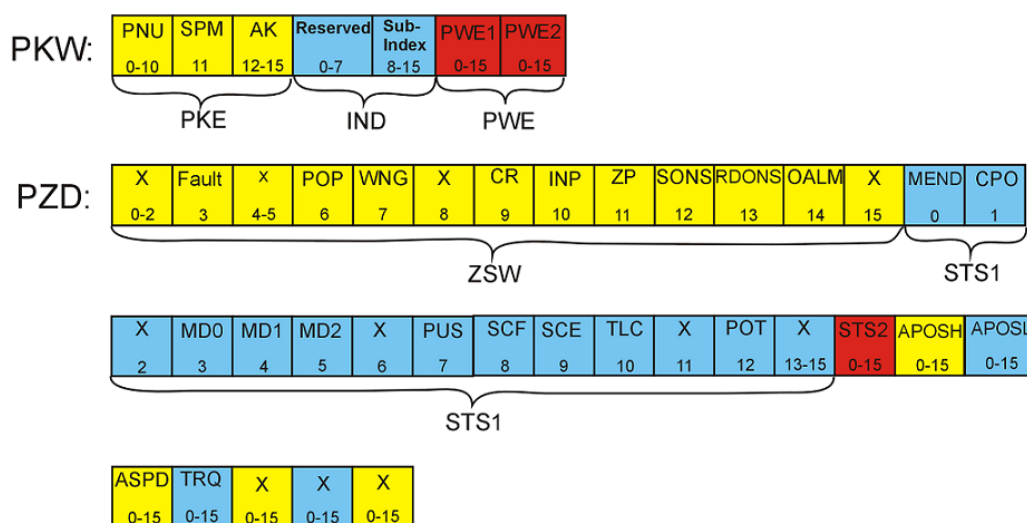


Obrázek 4.8 - Jednotlivá datová slova přenášeného datového rámce

Každý bit (0 – 15) přenášeného slova z datového rámce reprezentuje určitou přenášenou informaci, nebo úkon, jenž se má provést. Obsazení jednotlivých bitů příslušných slov popisují obrázky 4.9 a 4.10. Obrázek 4.9 popisuje jednotlivé bity datového rámce pro výstupní data (požadavek o příkaz) a obrázek 4.10 popisuje bity datového rámce pro vstupní data (odezva na příkaz).



Obrázek 4.9 - Obsazení jednotlivých bitů přenášeného datového rámce výstupních dat



Obrázek 4.10 - Obsazení jednotlivých bitů přenášeného datového rámce vstupních dat

Jak ilustrují předchozí obrázky, je v obou datových rámcích odlišné pouze bitové obsazení slov, jež jsou sdruženy jako PZD. Pro slova sdružená, jako PKW je obsazení bitů totožné jak pro datový rámec vstupních dat, tak i pro rámec výstupních dat [13]. Označení jednotlivých slov v datovém rámcu a použité názvy pro obsazení jednotlivých bitů příslušných slov jsou vysvětleny v příloze B.

Vzhledem k tomu, že jednotlivé operace které mají být provedeny, a které jednotka MR-MG30 řídí (směr rotace servomotoru, rychlost otáčení,...) se nastavují podle parametrů, je asi nejdůležitějším slovem z přenášeného datového rámce slovo PKE. Toto slovo nám totiž právě určuje charakteristiku parametru. Rozlišujeme 3 skupiny parametrů s nimiž jednotka MR-MG30 pracuje:

- Obecné parametry (1000 – 1059)
- Parametry osy
 - Parametry polohy (1200 – 1279)
 - Parametry servomotoru (1401 – 1439)

Obecné parametry – Nastavení parametrů, jenž jsou společné pro všechny osy a k nastavení parametrů pro komunikaci Profibus.

Parametry polohy – Nastavení jednotlivých os, dle požadavků aplikace a požadavků polohování.

Parametry servomotoru – Nastavení jednotlivých os, dle použitého servoměniče [13].

4.4 Programování a „oživení“ MR-MG30

Naprogramování jednotky MR-MG30 bylo provedeno prostřednictvím programovacího jazyka GX IEC Developer 7.00. Jedná se o programovací jazyk firmy Mitsubishi Electric, navržený speciálně pro programování takovýchto zařízení.

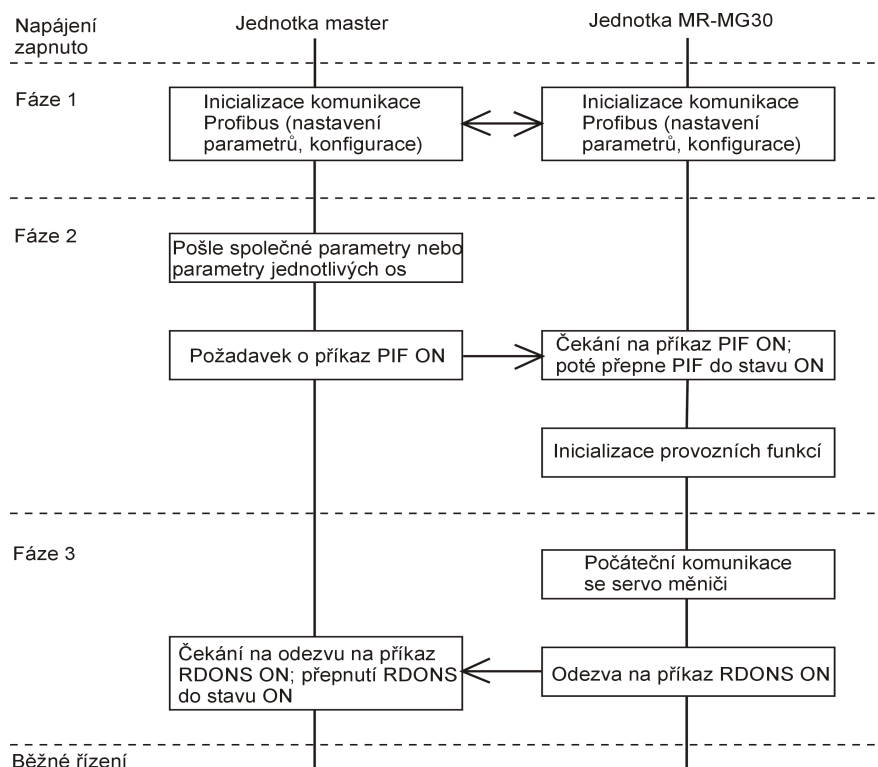
V jazyce GX IEC Developer rozdělujeme programování na programování v textovém editoru, nebo programování v grafickém editoru. V textovém editoru je možno psát vlastní program buďto jako klasický seznam instrukcí – IL (Instruction List), nebo jako strukturovaný text – ST (Structure Text). V případě seznamu instrukcí se jedná o jednoduchý textový editor a program se skládá ze sekvencí instrukcí, kde každá instrukce musí obsahovat jednu nebo několik proměnných. Dále musí každá instrukce obsahovat funkci, nebo operátor. Při tvorbě programu pomocí strukturovaného textu, se jedná o textově orientovaný programovací jazyk, jenž je podobný Pascalu a tudíž podporuje jednoduché matematické funkce. V grafickém editoru se program tvoří pomocí kontaktních schémat – LD (Ladder Diagram), diagramů funkčních bloků – FBC (Function Block Diagram), nebo pomocí sekvenčních funkčních diagramů – SFC (Sequential Function). V případě programu vytvořeného pomocí kontaktních schémat se program skládá z kontaktů, cívek, funkčních bloků a funkcí, přičemž funkční bloky a funkce jsou zde zobrazovány jako grafické bloky. Program vytvořen pomocí funkčních bloků je velice podobný kontaktnímu schématu, ale obsahuje pouze funkční bloky, případně funkce. Vlastní funkční bloky jsou vybírány z uživatelských knihoven a byly vytvořeny pomocí nějakého ze zmiňovaných způsobů programování, takže si funkční blok můžeme představit jako „masku“ jež nám ukryje program vytvořený pomocí jednoho ze zmiňovaných způsobů programování. Sekvenční funkční diagramy jsou tvořeny v grafickém programovacím jazyce. Vlastní program je tvořen pomocí dvou prvků – kroku a přechodu. Krok je buďto vstupní proměnná, nebo část programu vytvořená v jakémkoliv z výše zmiňovaných programovacích jazyků, přičemž může být vytvořen i přímo v editoru sekvenčních funkčních diagramů. Přechody jsou

naprogramovány pomocí seznamu instrukcí, kontaktních schémat, nebo editoru funkčních bloků. Každý přechod slouží jako přechodová podmínka mezi jednotlivými kroky a tudíž musí vždy nějakou podmínku obsahovat [17].

Vlastní program vytvořený v GX IEC Developeru 7.00 se skládá z jedné, nebo několika částí, vytvořených výše uvedenými způsoby. Každá tato část je označována jako programový modul – POU.

Před nahráním vlastního programu do paměti jednotky MR-MG30 je třeba ještě zajistit počáteční inicializaci jednotky. To bylo provedeno nahráním souboru MG3008D2.gsd do paměti jednotky pomocí softwaru GX Configurator DP. Tento soubor je dodáván od výrobce společně s MR-MG30 a zajišťuje uvedení do provozu jednotlivých funkcí a rysů jednotky – například maximální velikost vstupních a výstupních dat, dobu trvání pro přepínání mezi dvěma cykly, a podobně. Celý tento soubor je možné upravit v libovolném textovém editoru přesně pro potřeby uživatele. Poté co je do jednotky nahrán zmiňovaný konfigurační soubor a vlastní program do PLC, je MR-MG30 fyzicky připravena k práci. Po prvním spuštění dojde k inicializaci jednak vlastní komunikace mezi jednotlivými zařízeními, tak i k počáteční inicializaci připojených servoměničů a servomotorů. Počáteční inicializace je složena do třech fází. V první fázi dochází k nastavení komunikace Profibus na sběrnici RS-485 mezi jednotkou MR-MG30 a jednotkou typu master (PLC). Jednotka master vyšle sled konfiguračních příkazů pro každou připojenou osu. Dále vyšle informaci o počtu připojených os, čímž je dána velikost vysílaných/přijímaných dat. V následující fázi inicializačního procesu obdrží jednotka MR-MG30 od jednotky master data PKW, a tím je zajištěno předání parametrů společných pro všechny osy. Dále pokud je to nezbytné, vyšle jednotka master do MR-MG30 ještě parametry, které jsou již pro každou osu specifické. Po zaslání všech potřebných parametrů jednotka master vyšle ještě požadavek na příkaz PIF (Parameter initial transfer completed) ON. Tím je zajištěno, že MR-MG30 po přijetí všech parametrů jednotlivých os přepne příznak PIF do stavu ON a inicializační proces může přejít do třetí fáze. V poslední fázi inicializačního procesu jednotka MR-MG30 nastaví a uloží jednotlivé parametry, přijaté ve druhé fázi. Poté naváže počáteční komunikaci s jednotlivými servoměniči (podle počtu nastavených os v první fázi) a po dokončení vyšle jednotce master informaci o tom, že bylo vše úspěšně

provedeno – nastavením příznaku RDONS (Servo amplifier control ready) do stavu ON (odezva na příkaz). Jednotka Master po přijetí tohoto příznaku ukončí poslední fázi inicializačního procesu a přepne na běžné řízení [13]. Celý inicializační proces ilustruje obrázek 4.11.



Obrázek 4.11 - Inicializační sekvence MR-MG30

Při úplném prvním spuštění jednotky MR-MG30 dochází ve třetí fázi k zmiňované komunikaci s jednotlivými připojenými servoměniči. Při této komunikaci dochází k nahrání parametrů příslušné osy do odpovídajícího servoměniče. Při opakovaných spuštěních se již toto neodehrává, jelikož servoměnič si potřebné informace uchovává ve vlastní paměti. Proto při opakovaných spuštěních pouze jednotka master obdrží informaci o tom, že jsou jednotlivé servoměniče a tím pádem i připojené servomotory připraveny k práci. Při inicializačním procesu jednotlivé servoměniče zobrazují aktuální stav na dvojici svých sedmi-segmentových displejů prostřednictvím kombinací číslic a písmen. Každá kombinace představuje právě vykonávanou práci, nebo případnou chybu inicializace. Pokud zobrazovaný údaj obsahuje číslici, je jejím prostřednictvím předávána informace o čísle osy, se kterou servoměnič pracuje. Po proběhnutí

```

graph TD
    Start([Servo měnič zapnut]) --> Ab[A b]
    Ab -- "Řadič servo systému zapnut" --> AC[A C]
    AC --> Ad[A d]
    Ad --> Ae[A E]
    Ae --> b1[b 1]
    b1 --> C1[C 1]
    C1 --> d1[d 1]
    d1 --> Bp[Běžný provoz]
    Bp --> Start
    Bp -- "Pokud nastane alarm, objeví se jeho kód" --> b1
    Bp -- "Pokud nastane alarm, objeví se jeho kód" --> C1
    Bp -- "Pokud nastane alarm, objeví se jeho kód" --> d1
    b1 --> E7[E 7]
    b1 --> E6[E 6]
    E7 -- "nebo" --> E6
    E6 -- "Reset nouzového zastavení a vynuceného zastavení" --> Start
    Bp -- "Řadič servo systému vypnut" --> AA[A A]
    AA -- "Řadič servo systému zapnut" --> Start
  
```

Obrázek 4.12 - Inicializační sekvence servoměniče

Jednotlivé možné kombinace, které může servoměnič zobrazovat během práce na displej jsou popsány v tabulce 4.5 – viz [18]. Znak # v tabulce, jemuž ještě předchází nějaké písmeno (např. d), označuje připojenou osu k servoměniči – například, pokud se na displeji servoměniče objeví dvojice znaků d 1, servoměnič nám oznamuje, že připojená osa 1 (servomotor) je zapnuta a připravena k práci. Dvojice znaků * * v tabulce představuje nějaký alarm nebo hlášení (warning) – pokud nějaký alarm nebo hlášení nastane, bude místo " * *" zobrazena nějaká dvojice čísel, která představuje

daný alarm nebo hlášení (warning).

Údaj	Stav	Popis
A A	Inicializace	Servoměnič byl zapnut; předtím byl řadič servosystému vypnut
A b	Inicializace	<ul style="list-style-type: none"> - Napájení řadiče servosystému bylo vypnuto během zapínání servoměniče. - Číslo osy nastavené na řadiči servosystému nesouhlasí s číslem osy nastavené na přepínači servoměniče. - Nastala chyba servoměniče, nebo nastala chyba komunikace s řadičem servosystému. V tomto případě je posloupnost znaků následující: „Ab“ -> -> „AC“ -> „Ad“ -> „Ab“. - Vadný řadič servosystému.
A C	Inicializace	Začala komunikace mezi řadičem servosystému a servoměničem.
A d	Inicializace	Počáteční parametry z řadiče servosystému byly přijaty.
A E	Ukončení inicializace	Počáteční datová komunikace s řadičem servosystému byla dokončena.
b # (Pozn. 1)	Příznak Ready ve stavu OFF	Byl přijat signál Ready OFF z řadiče servosystému.
d # (Pozn. 1)	Servo ON	Byl přijat signál Servo ON z řadiče servosystému.
C # (Pozn. 1)	Servo OFF	Byl přijat signál Servo OFF z řadiče servosystému.
* * (Pozn. 2)	Alarm / Warning	Zobrazení čísla Alarmu / Warningu, který nastal.
8 8	Chyba CPU	Chyba CPU řadiče servosystému.
b 0 (Pozn. 3)	Test pracovního režimu	Režim JOG, režim polohování, naprogramovaný režim práce, vynucený výstup DO (Digital Output).
b # (Pozn. 1)	Test pracovního režimu	Bezmotorový režim.
d # (Pozn. 1)	Test pracovního režimu	Bezmotorový režim.
c # (Pozn. 1)	Test pracovního režimu	Bezmotorový režim.

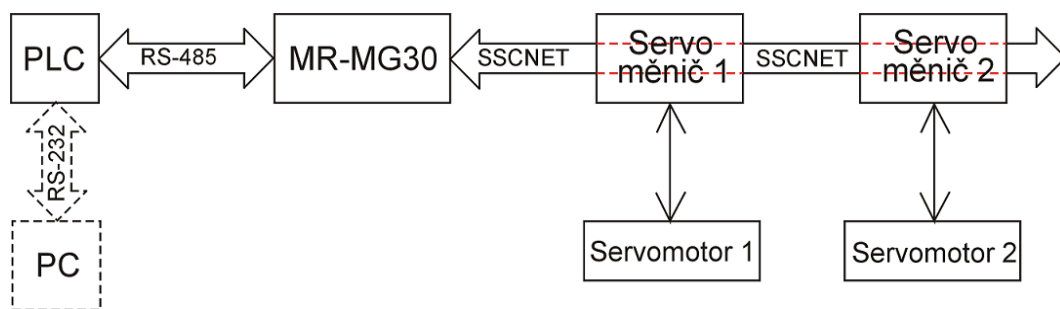
Tabulka 4.5 - Možné zobrazené kombinace na displeji servoměniče

5 Model dvouosého lineárního servopohonu

Modelová úloha dvouosého servopohonu byla postavena za účelem seznámení se s prací a programováním řídicí jednotky MR-MG30.

5.1 Návrh a realizace modelu

Pro realizaci modelu byly zvoleny komponenty od firmy Mitsubishi Electric, z důvodu kompatibility s řídicí jednotkou MR-MG30 a díky specifickému již zmiňovanému rozhraní SSCNET II. Model je sestaven z PLC (jednotka master), k němuž je prostřednictvím RS-485 připojena řídicí jednotka MR-MG30. K PLC může být ještě připojeno PC (za účelem nahrání ovládacího programu do PLC) prostřednictvím rozhraní RS-232. K jednotce MR-MG30 je skrze rozhraní SSCNET připojena dvojice servoměničů a ke každému z nich je připojen trojfázový střídavý rotační servomotor. Blokové schéma modelu je na obrázku 5.1.



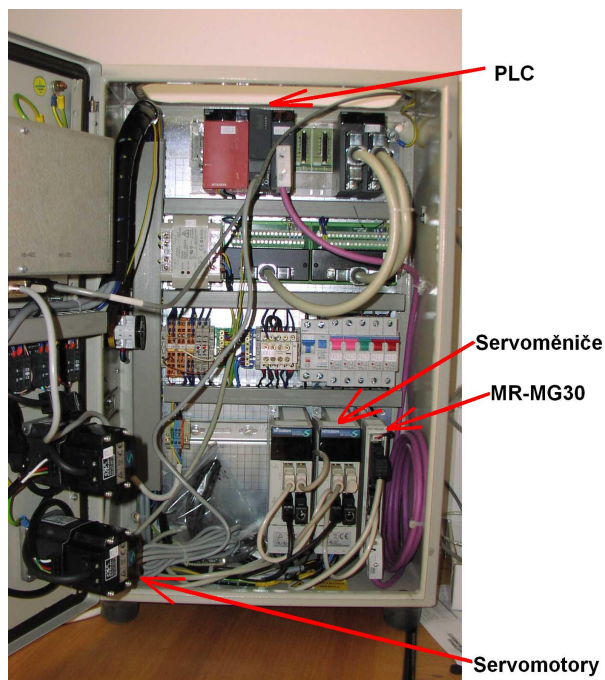
Obrázek 5.1 - Blokové schéma dvouosého lineárního servopohonu

V PLC je nahrán řídicí program celého modelu dvouosého servopohonu. Prostřednictvím panelu PLC jsou ovládány jednotlivé osy servopohonu a dále jsou na panel zobrazovány informace o jednotlivých stavech modelu – např. chybová hlášení,

informace o aktuální poloze jednotlivých os atd. Řídicí jednotka MR-MG30 má za úkol zprostředkovat komunikaci mezi PLC a připojenými servoměniči a dále předávání parametrů jednotlivých os servoměničům. Dvojice servoměničů pak zajišťuje ovládání a komunikaci s připojenými servomotory. Model dvouosého servopohonu je na obrázku 5.2 a vnitřní uspořádání modelu ilustruje obrázek 5.3.



Obrázek 5.2 - Model dvouosého servopohonu



Obrázek 5.3 - Vnitřní uspořádání modelu dvouosého servopohonu

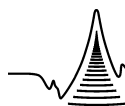
5.1.1 Parametry jednotlivých komponent modelu

Jak již bylo zmíněno, jako jednotka master celého modelu slouží PLC. Jedná se o modulární víceprocesorové PLC. PLC je osazeno procesorem Q02CPU a lze připojit až čtyři samostatné CPU, které se podílejí na řízení zátěže, řízení pohybu a komunikaci s PC. Parametry tohoto procesoru popisuje tabulka 5.1 – viz [19].

Typ	Víceprocesorový CPU modul
Počet vstupů / výstupů	4096 / 8192
Samodiagnostické funkce CPU	Detekce chyb CPU, watch dog, detekce chyb baterie, detekce chyb paměti, kontrola programu, detekce chyb napájení
Baterie	Lithiová baterie
Typy paměti	RAM, ROM, Flash
Kapacita paměti	≤ 32 MByte
Paměť pro program	28000 kroků
Doba vykonání instrukce	79 ns

Tabulka 5.1 - Parametry procesoru Q02CPU

V případě použitých servoměníčů, se jedná o dva shodné servoměníče MR-J2S-20B. Tyto servoměníče umožňují ovládat servomotory až do celkového výkonu 200 W. K servoměníčům je připojena dvojice shodných servomotorů HC-KFS23, což jsou servomotory o maximálním výkonu 200W. Vlastnosti servoměníčů popisuje tabulka 5.2 a vlastnosti použitých servomotorů popisuje tabulka 5.3 – více v [18].



Napětí /	3-fázové, 200 – 230 V AC; 1-fázové 230 V AC
Frekvence	50/60 Hz pro 3-fázové napájení 50/60 Hz pro 1-fázové napájení
Přípustné kolísání napětí	175V – 253V AC pro 3-fázové napájení 207V – 253V AC pro 1-fázové napájení
Přípustné kolísání frekvence	± 5 %
Řídicí systém	Sinusově řízená PWM, proudově řízený systém
Ochranné funkce	Proudová pojistka, přepětíová pojistka, ochrana proti přetížení motoru, ochrana proti přehřátí motoru, ochrana proti výpadku napájení, ochrana proti překročení maximálních otáček motoru, ochrana proti podpětí
Kapacita napájecího zdroje	0,5 kVA
Zapínací proud	30 A pro hlavní obvod 70 – 100 A pro regulační obvod

Tabulka 5.2 - Základní parametry servoměniče MR-J2S-20B

Výkon napájecího zdroje [kVA]	0,5
Jmenovitý výkon [W]	200
Jmenovitý točivý moment [Nm]	0,64
Maximální točivý moment [Nm]	1,9
Jmenovité otáčky [ot/min]	3000
Maximální otáčky [ot/min]	4500
Přípustné okamžité otáčky [ot/min]	5175
Jmenovitý proud [A]	1,5
Maximální proud [A]	5,0
Moment setrvačnosti J [$\times 10^{-4}$ kgm ²]	0,088
Počet pulsů na jednu otáčku [p/ot]	131072
Doporučená zátěž/moment setrvačnosti	≤ 15x menší hodnota, než moment setrvačnosti

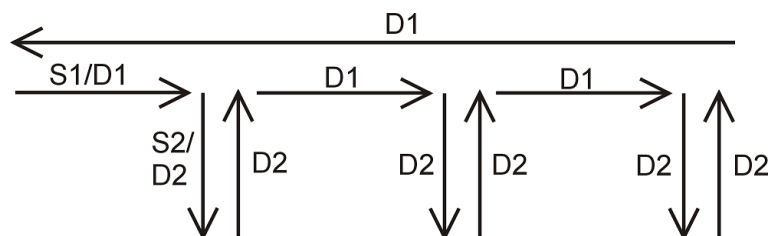
Tabulka 5.3 - Základní parametry servomotoru HC-KFS23

5.2 Program ovládající model

Řídicí program pro ovládání dvouosého lineárního servopohonu byl vytvořen v programovacím jazyce od firmy Mitsubishi Electric GX IEC Developer 7.00. Při

konstrukci programu jsem vycházel z výukového programu pro ovládání servopohonů s využitím řídicí jednotky MR-MG30, jenž byl vytvořen firmou Beijer Electronic AB. Tento program byl vytvořen za účelem pochopení a získání základního povědomí o programování řídicí jednotky MR-MG30 a byl dodán společně s jednotkou MR-MG30.

Vlastní program pro ovládání modelu dvouosého servopohonu jsem vytvořil pomocí editoru funkčních bloků FBC a je rozdělen do čtyřech částí (podprogramů). Jednotlivé podprogramy jsou tedy realizovány jako funkční bloky FB, které byly vytvořeny v editoru kontaktních schémat LD. Program byl navržen jako manipulátor s výrobky, takže jedna osa vykonává horizontální pohyb a druhá osa vykonává pohyb vertikální. Polohování jednotlivých os servopohonu ilustruje obrázek 5.4

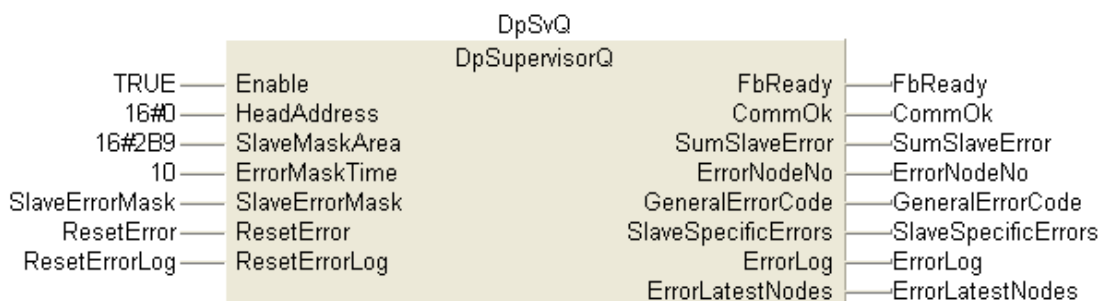


Obrázek 5.4 - Schéma polohování jednotlivých os servopohonu

Jak je patrné z obrázku (Obr. 5.4), šipky představují směr pohybu servopohonu a jednotlivé symboly představují příznaky, jimiž se uvede do provozu polohování příslušné osy. Po prvním spuštění program čeká na nastavení příznaku S1, čímž dojde k zahájení polohování 1.osy servopohonu do požadované polohy. Po dosažení cílové polohy 1.osy opět program čeká na nastavení příznaku pro 2.osu servopohonu S2. Po jeho nastavení se zahájí polohování 2.osy servopohonu do požadované polohy. Jakmile 2.osa servopohonu dosáhne požadované polohy, program čeká na nastavení příznaku D2 a po jeho nastavení opět polohuje 2.osa servopohonu do příslušné požadované polohy. Celý tento postup se zopakuje ještě dvakrát a poté se 1.osa po nastavení příznaku D1 vrací do polohy, která byla definována jako nulová (Position0 – viz obr. 5.6). Po návratu 1.osy servopohonu se celý běh bude opět cyklicky opakovat, ale s tím rozdílem, že místo příznaků S1 a S2 – viz obr. 5.4 program čeká na nastavení příznaků D1 a D2. Pokaždé když servopohon zahajuje polohování příslušné osy, je právě aktivní příznak

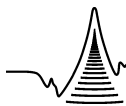
opět resetován, aby mohl být následně opět použit k spuštění dalšího polohování. Příznak S1 (S2) představuje zahájení polohování příslušné osy servopohonu, kdežto příznak D1 (D2) představuje polohování do následující polohy servopohonu po již zahájení polohování (již byly aktivovány příznaky S1 a S2 po prvním spuštění programu). Pokud by došlo k vypnutí jednoho, nebo obou servomotorů a k jejich následnému spuštění, bude se celý program opakovat od svého začátku – bude se tedy muset opět nejprve spustit polohování příslušných os pomocí příznaků S1 a S2 a po jejich aktivaci budou jednotlivé osy opět ovládány příznaky D1 a D2 pro polohování do následujících požadovaných poloh. Jednotlivé příznaky jsou ovládány pomocí tlačítek na ovládacím panelu PLC. Příznak S1 se aktivuje tlačítkem *Start poloh.* a všechny ostatní příznaky (S2, N1, N2) jsou aktivovány tlačítkem *Dalsi poloha* – viz kapitola 5.2.1.

První funkční blok *DPSupervisorQ*, byl převzat z již zmiňovaného programu pro pochopení programování MR-MG30. Blok *DPSupervisorQ* zajišťuje komunikaci prostřednictvím standardu Profibus mezi připojeným PLC a řídicí jednotkou MR-MG30. Vzhled tohoto funkčního bloku ilustruje obrázek 5.5.



Obrázek 5.5 - Funkční blok *DPSupervisorQ*

Jednotlivé vstupní a výstupní proměnné funkčního bloku *DPSupervisorQ* popisuje tabulka 5.4.

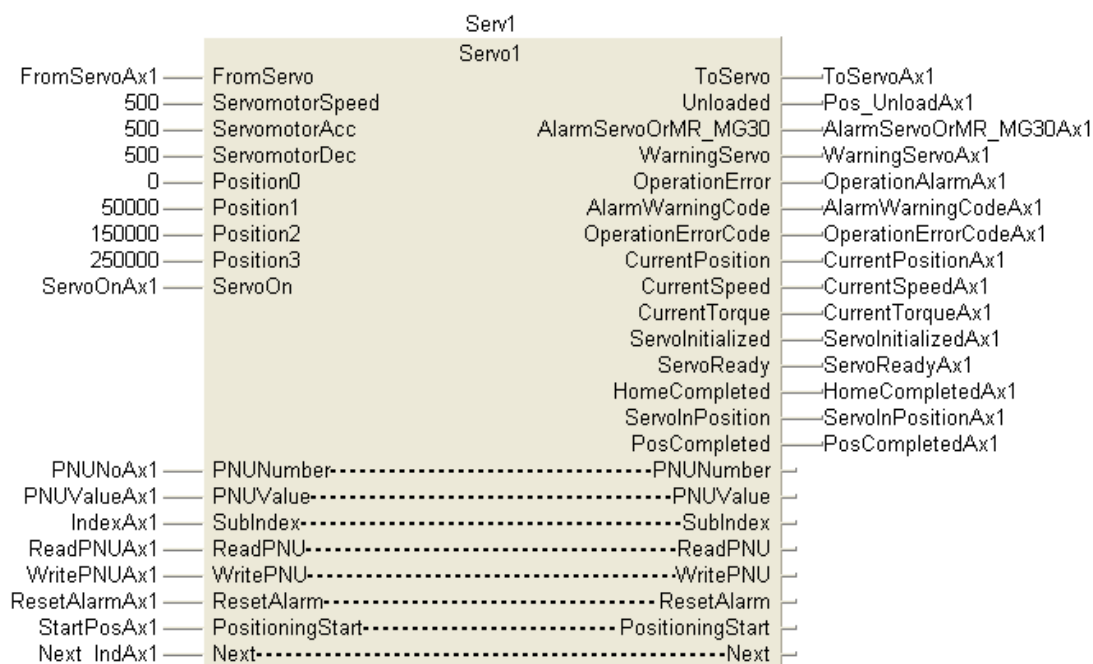


Vstupní proměnné	Popis
Enable	Musí být true, jinak nebude funkční blok použit
HeadAddress	Adresa na PLC
SlaveMaskArea	Oblast pro zrušení informací o problémech se zařízením slave
ErrorMaskTime	Čas, během kterého nejsou zobrazovány komunikační chyby
SlaveErrorMask	Přiřazení čísla (0-126) příslušnému alarmu
ResetError	Reset kódu chyby z bufferu paměti
ResetErrorLog	Reset záznamu chyby z paměti
Výstupní proměnné	Popis
FbReady	Start komunikace
CommOk	Příznak, že nenastal žádný problém s komunikací na Profibusu
SumSlaveError	Příznak, že nenastal žádný problém s komunikací na Profibusu
ErrorNodeNo	Proměnná udávající informaci o jakou chybu se jedná (číslo 0-126)
GeneralErrorCode	Oblast pro problémy s komunikací
SlaveSpecificErrors	Oblast rozšiřující oblast pro problémy s komunikací – rozšířeno o 8 specifických chyb
ErrorLog	Číslo alarmu
ErrorLatestNodes	Proměnná obsahující 8 posledních chyb

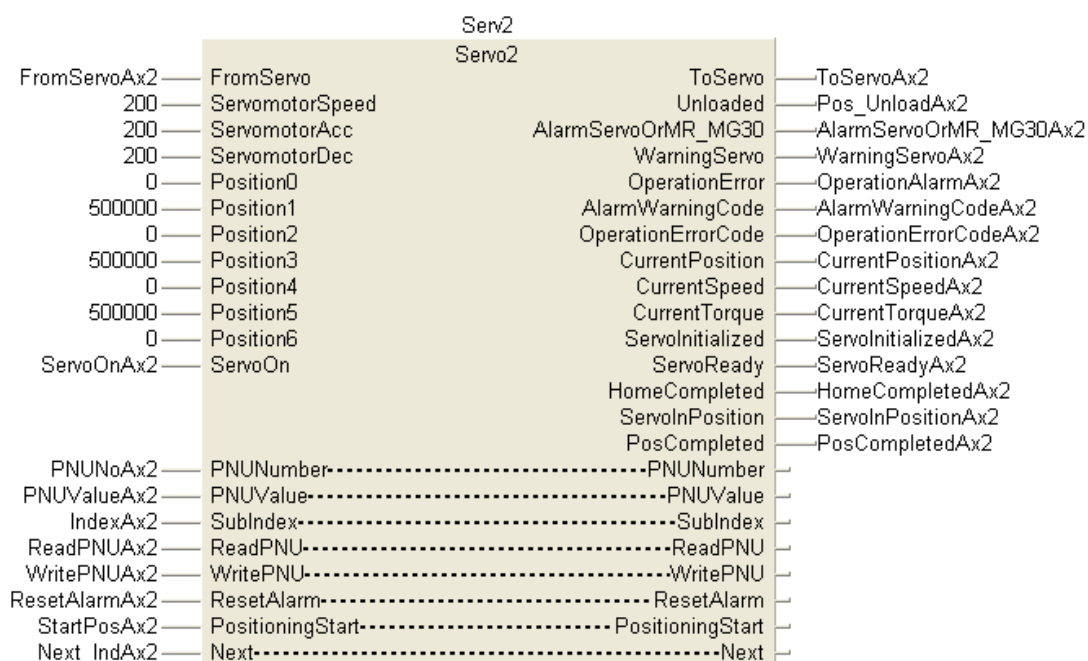
Tabulka 5.4 - Význam jednotlivých proměnných funkčního bloku DPSSupervisorQ

Následující dvojice funkčních bloků *Servo1* a *Servo2* byla vytvořena pro ovládání jednotlivých os servopohonu. Prostřednictvím těchto funkčních bloků se zadává požadovaná rychlost a požadované časové konstanty zrychlení a zpomalení servomotoru. Dále se prostřednictvím dvojice funkčních bloků *Servo1* a *Servo2* zadávají požadované polohy, do kterých budou jednotlivé servomotory postupně polohovat. Tato dvojice funkčních bloků také zajišťuje spuštění servomotoru, pokud obdrží požadavek (vstupní signál) od funkčního bloku *Control*, jenž zajišťuje koordinaci práce obou os servopohonu – viz dále. Funkční bloky *Servo1* a *Servo2* musí dále vyhodnotit, zda nedošlo k nějaké chybě, ať už komunikace mezi PLC a servoměničem, nebo provozní chybě servomotoru. Pokud nějaká z těchto chyb, nebo hlášení nastane, funkční blok nastaví příslušný příznak a do příslušné proměnné (viz tab. 5.5) uloží kód chyby, jež nastala. PLC následně zobrazí kód chyby a její název na ovládací panel. Pomocí ovládacího panelu PLC pak může obsluha po vyhodnocení příslušné chyby a jejím odstranění provést reset příslušných příznaků a proměnných funkčního bloku. Funkční blok poté znovu vyhodnotí, zda nenastala nějaká chyba a pokud ne, může příslušná osa

servopohonu pokračovat v práci. Dále je možné pomocí těchto funkčních bloků přepsat, nebo pouze přecházet hodnoty jednotlivých parametrů servomotoru – viz [13] (maximální točivý moment, směr otáčení,...), pokud by například došlo k výměně servomotorů za jiné. Veškeré požadavky funkčního bloku na servoměnič a jeho prostřednictvím i na servomotor jsou posílány pomocí posloupnosti čtrnácti datových slov – proměnná ToServo a jejich prostřednictvím opět funkční blok přijímá data od servoměniče – proměnná FromServo. Datová slova byla popsána v kapitole 4.2.1. Dvojice funkčních bloků *Servo1* a *Servo2* byla vytvořena s naprosto shodnou konstrukcí a liší se pouze v počtu vstupních proměnných, jež zajišťují načtení požadovaných poloh, do kterých bude servomotor postupně polohovat. Tuto dvojici funkčních bloků ilustrují obrázky 5.6 a 5.7.

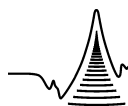


Obrázek 5.6 - Funkční blok Servo1



Obrázek 5.7 - Funkční blok Servo2

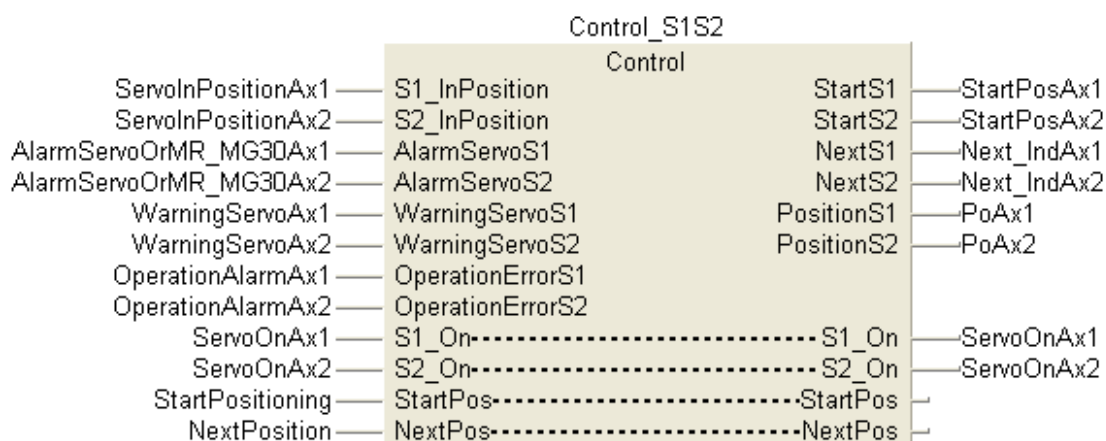
Jak je patrné z obou obrázků (obr 5.5 a 5.6), mají funkční bloky také rozdílné vstupní a výstupní signály – globální proměnné, které vstupují do funkčního bloku, nebo z něho vystupují ven. Globální proměnné jsou nastavovány prostřednictvím vstupních, vstupně-výstupních nebo výstupních proměnných. Pomocí těchto signálů (globálních proměnných) spolu komunikují jednotlivé funkční bloky a jejich prostřednictvím si předávají příslušné informace a parametry. Dalším významem těchto signálů je zajištění komunikace řídicího programu PLC s programem ovládacího panelu PLC – viz kapitola 5.2.1. Jednotlivé vstupní, vstupně-výstupní a výstupní proměnné funkčních bloků *Servo1* a *Servo2* popisuje tabulka 5.5.



Vstupní proměnné	Popis
FromServo	Posloupnost 14 řídicích slov (16-ti bitových) přijímaných od servoměniče
ServomotorSpeed	Požadovaná rychlost servomotoru
ServomotorAcc	Požadovaná časová konstanta zrychlení servomotoru
ServomotorDec	Požadovaná časová konstanta zpomalení servomotoru
Position0	Výchozí poloha servomotoru
Position1 ⋮ Position6	Požadované polohy servomotoru
ServoOn	Příznak, že má dojít k zapnutí servomotoru
Vstupně-výstupní proměnné	Popis
PNUNumber	Číslo parametru, který se má číst, nebo přepsat
PNUValue	Hodnota požadovaného parametru-proměnná se používá pro zápis, nebo slouží pro zobrazení při čtení
SubIndex	Pro zápis, nebo čtení musí být 0; jinak slouží jako SubIndex – viz [13]
ReadPNU	Příznak, že se má požadovaný parametr číst
WritePNU	Příznak, že se má požadovaný parametr přepsat
ResetAlarm	Reset alarmu / warningu, jenž nastal
PositioningStart	Start polohování
Next	Po startu polohování servo čeká na aktivaci tohoto příznaku, pro polohování do následující polohy
Výstupní proměnné	Popis
ToServo	Posloupnost 14 řídicích slov (16-ti bitových), jenž jsou posílány do servoměniče
Unloaded	Příznak, že požadované polohy byly načteny
AlarmServoOrMR_MG30	Příznak, že nastala chyba servomotoru, nebo jednotky MR-MG30
WarningServo	Příznak, že nastalo nějaké hlášení servomotoru
OperationError	Příznak, že nastala nějaká provozní chyba
AlarmWarningCode	Kód příslušné chyby nebo hlášení, jenž může nastat
OperationErrorCode	Kód příslušné provozní chyby, jenž může nastat
CurrentPosition	Aktuální poloha servomotoru
CurrentSpeed	Aktuální rychlost servomotoru
CurrentTorque	Aktuální točivý moment servomotoru
ServoInitialized	Příznak, že inicializace servoměniče nastala bez chyby
ServoReady	Příznak, že je servomotor připraven k práci
HomeCompleted	Příznak, že servomotor vrátil nulovou polohu
ServoInPosition	Příznak, že je servomotor v požadované poloze
PosCompleted	Příznak, že polohování bylo dokončeno

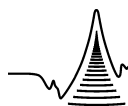
Tabulka 5.5 - Význam jednotlivých proměnných funkčních bloků Servo1 a Servo2

Posledním funkčním blokem, který byl vytvořen je blok *Control*. Funkční blok *Control* zajišťuje koordinaci práce a ovládání dvojice funkčních bloků *Servo1* a *Servo2*. Jeho úkolem je zajistit spouštění příslušné osy servopohonu podle již zmiňovaného schématu polohování jednotlivých os servopohonu (obr. 5.4). Funkční blok *Control* ilustruje obrázek 5.8.



Obrázek 5.8 - Funkční blok Control

Jednotlivé vstupní, vstupně-výstupní a výstupní proměnné funkčního bloku *Control* popisuje tabulka 5.6.



Vstupní proměnné	Popis
S1_InPosition	Příznak, že je 1.osa servopohonu v požadované pozici
S2_InPosition	Příznak, že je 2.osa servopohonu v požadované pozici
AlarmServoS1	Příznak, že nastala nějaká chyba 1.osy servopohonu
AlarmServoS2	Příznak, že nastala nějaká chyba 2.osy servopohonu
WarningServoS1	Příznak, že nastalo nějaké hlášení 1.osy servopohonu
WarningServoS2	Příznak, že nastalo nějaké hlášení 2.osy servopohonu
OperationErrorS1	Příznak, že nastala nějaká provozní chyba 1.osy servopohonu
OperationErrorS2	Příznak, že nastala nějaká provozní chyba 2.osy servopohonu
Vstupně výstupně proměnné	Popis
S1_On	Proměnná, která zapne 1.osu (servomotor) servopohonu
S2_On	Proměnná, která zapne 2.osu (servomotor) servopohonu
StartPos	Proměnná, která spustí polohování servopohonu
NextPos	Příznak, že má servopohon polohovat do další požadované polohy
Výstupní proměnné	Popis
StartS1	Příznak, že se má spustit polohování servomotoru 1
StartS2	Příznak, že se má spustit polohování servomotoru 2
NextS1	Příznak, že má 1.osa servopohonu polohovat do další požadované polohy
NextS2	Příznak, že má 2.osa servopohonu polohovat do další požadované polohy
PositionS1	Proměnná udávající, ve které pozici se nachází servomotor 1
PositionS2	Proměnná udávající, ve které pozici se nachází servomotor 2

Tabulka 5.6 - Jednotlivé proměnné funkčního bloku Control

Jednotlivé globální proměnné, které vstupují nebo vystupují z jednotlivých funkčních bloků jsou vysvětleny v příloze C.

5.2.1 Program pro ovládací panel PLC

Program pro ovládací panel PLC byl vytvořen pomocí software E-Designer 7 od firmy Mitsubishi Electric. Více o tomto software a o programování ovládacího panelu PLC jeho prostřednictvím se lze dočíst v [20]. V tomto prostředí bylo navrženo

uspořádání jednotlivých ovládacích prvků a prvků zajišťujících informace o jednotlivých stavech servopohonu. Jak již bylo zmíněno v kapitole 5.2, informace od řídicího programu PLC pro program ovládacího panelu PLC a opačně, jsou předávány prostřednictvím globálních proměnných. Každá z těchto proměnných je uložena v nějakém registru PLC. V našem případě jsou použity pouze datový registr D (data register) a vnitřní přenos M (internal relay). Datový registr D je využíván pro předávání např. informací o aktuální poloze servomotoru, pro předávání hodnot jednotlivých parametrů servomotoru, nebo pro předávání jednotlivých řídicích datových slov. Proměnné uložené v datovém registru D jsou typu int, dint nebo array of int. Vnitřní přenos M slouží pro ovládací funkce např. zapnutí nebo vypnutí servomotoru, nebo pro zobrazení jednotlivých stavů servomotoru např. servo je připraveno k práci. Globální proměnné využívající registru M jsou datového typu bool. Jednotlivé proměnné poté využívají jednotlivé ovládací prvky a prvky informativního charakteru, s ohledem na jejich datové typy.

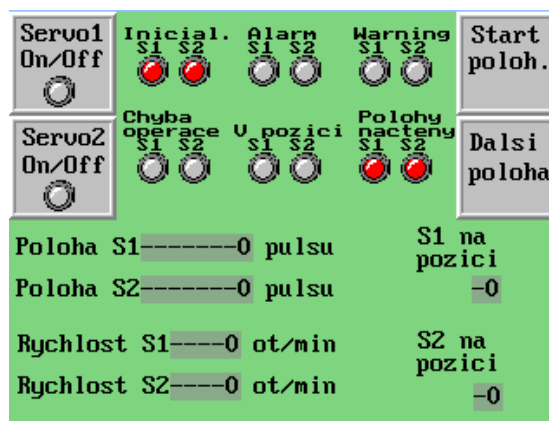
Program pro ovládací panel PLC byl vytvořen jako čtveřice programových bloků, kde každý blok představuje jednu obrazovku, na kterou se může obsluha PLC postupně přepnout. Každá obrazovka je tvořena různými komponentami, jimž je přiřazena konkrétní globální proměnná. Podle datového typu globální proměnné je pak určeno, k čemu daná komponenta bude sloužit, jak již bylo zmiňováno.

Prvním vytvořeným programovým blokem je blok *Main*, který je úvodní obrazovkou ovládacího panelu PLC po jeho zapnutí. Jednotlivá tlačítka na této obrazovce slouží pouze k přepnutí se na následující obrazovky, které reprezentují jednotlivé programové bloky. Ovládací panel PLC s uspořádáním úvodní obrazovky, která reprezentuje programový blok *Main* ilustruje obrázek 5.9.



Obrázek 5.9 - Ovládací panel PLC s úvodní obrazovkou

Následujícím programovým blokem je blok *Polohovani*, jenž zajišťuje ovládání jednotlivých os servopohonu. Rozložení jednotlivých prvků na obrazovce, která reprezentuje programový blok *Polohovani* ilustruje obrázek 5.10.



Obrázek 5.10 - Obrazovka reprezentující programový blok Polohování

Tlačítka *Servo1 (2) On/Off* slouží k zapnutí, nebo vypnutí příslušné osy servopohonu. Tlačítko *Start poloh.* je určeno ke spuštění polohování servopohonu – nastaví se příznak

S1 -viz kapitola 5.2. Spuštění polohování je možné provést až když jsou obě osy servopohonu zapnuty. Pokud by došlo k vypnutí jedné z os servopohonu a jejímu následnému spuštění, musí se opět k zahájení polohování toto tlačítko použít. Tlačítko *Dalsi poloha* je určeno k polohování servopohonu do následující pozice servopohonu dle schématu polohování (obr. 5.4). Tlačítko *Dalsi poloha* může představovat nějaké koncové spínače, jenž by na tento vstup byly připojeny, a tím dávaly informaci, že servopohon může přejít do další polohy. Dále jsou na této obrazovce zobrazovány jednotlivé stavy servomotorů, informace že nastala chyba, nebo chybové hlášení a v neposlední řadě jsou zde zobrazeny informace o aktuálních rychlostech a pozicích servomotorů.

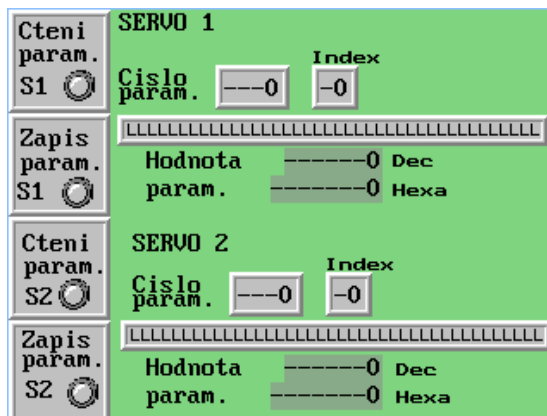
Další obrazovka reprezentuje programový blok *Alarmy*. Obrazovka reprezentující programový blok *Alarmy* slouží pro zobrazení jednotlivých chyb, nebo chybových hlášení ať už komunikace mezi řídicí jednotkou MR-MG30 a PLC, nebo provozní chyby servomotorů. Obrazovku programového bloku *Alarmy* ilustruje obrázek 5.11.



Obrázek 5.11 - Obrazovka reprezentující programový blok Alarmy

Na této obrazovce (obr. 5.11) lze zapnout jednotlivé servomotory pomocí tlačítek *Servo1(2) On/Off* a dále provést reset jednotlivých chyb, nebo chybových hlášení tlačítkem *Reset alarm S1 (S2)*, poté co se chyba / chybové hlášení zobrazí. Zbývající komponenty na obrazovce podávají informace o kódu a názvu chyby, nebo chybového hlášení.

Poslední obrazovkou která byla vytvořena, je obrazovka reprezentující programový blok *Parametry*. Prostřednictvím této obrazovky je možné přepisovat, nebo číst hodnoty jednotlivých parametrů servomotorů – viz [13]. Obrazovka reprezentující programový blok *Parametry* ilustruje obrázek 5.12.



The screenshot shows a software interface for configuring two servomotors, SERVO 1 and SERVO 2. For each motor, there are two main sections: 'Cteni param.' (Read parameter) and 'Zapis param.' (Write parameter). Each section contains a button labeled 'S1' or 'S2', a field for 'Cislo param.' (Parameter number) with a value of 0, and a field for 'Index' with a value of 0. Below these is a large display area for 'Hodnota param.' (Parameter value), which can be shown in 'Dec' (decimal) or 'Hexa' (hexadecimal) format, with a value of 0 displayed.

Obrázek 5.12 - Obrazovka reprezentující programový blok Parametry

Tlačítko *Cteni param. S1 (S2)* zajistí zobrazení hodnoty parametru, po zadání jeho čísla do pole označeného jako *Cislo param.* Následně pak bude zobrazen název příslušného parametru servomotoru a hodnota bude zobrazena jak v dekadickém, tak i v hexadecimálním tvaru prostřednictvím komponent označených jako *Hodnota param.* Pokud chceme nějaký parametr servomotoru přepsat, musí se opět zadat číslo požadovaného parametru, jako při požadavku na jeho četní. Poté do polí označených jako *Hodnota param.* zapíšeme hodnotu parametru buďto v dekadickém nebo hexadecimálním tvaru a nakonec stiskneme tlačítko *Zapis param. S1 (S2)*. Tím je hodnota příslušného parametru natrvalo uložena.

Závěr

Hlavním cílem této diplomové práce bylo prostudování vlastností řídicí jednotky MR-MG30 od firmy Mitsubishi Electric a její následné využití pro řízení servomotorů v modelu dvouosého servopohonu. Pro jednodušší aplikace je řídicí jednotka MR-MG30 plně dostačující. Jednotka nabízí velmi zajímavé řešení integrace servomotorů připojovaných přes rozhraní SSCNET na sběrnici Profibus a tím i velmi široké použití ve výrobě, neboť nabídka servomotorů, jež se tímto rozhraním připojují, je velice široká a tyto servomotory mají celou řadu výhod – vysoká dynamika, funkce pro automatické nastavování v reálném čase apod. Jednotka MR-MG30 sice umožňuje řízení až šesti os servopohonu, avšak dokáže řídit vždy jen jednu osu a tím pádem není schopna řídit více os najednou. Z tohoto důvodu by se pro složitější aplikace muselo hledat jiné řešení.

Ovládací program pro PLC byl vytvořen v softwaru GX IEC Developer 7.00, ale možná by bylo vhodnější program vytvořit například v jazyce C, z důvodu přehlednosti programu. Ovšem otázkou zůstává, zda by při použití jazyka C bylo možné program ladit stejně dobře, jako v softwaru GX IEC Developer 7.00, který umožňuje přepnutí do takzvaného transparentního módu, ve kterém je možné sledovat jaké signály vstupující, nebo vystupující do programu jsou právě aktivní a jaké hodnoty jsou uloženy v PLC. Další výhodou programování v GX IEC Developer 7.00, oproti tvorbě programu například v jazyce C je, že tento software byl navržen speciálně pro programování PLC, tudíž má již v sobě uloženu knihovnu obsahující jednotlivé série PLC s odpovídajícími typy procesorů. Tím pádem stačí pouze zvolit požadovanou sérii a procesor a dále se programátor již nemusí starat o specifika kompilace na konkrétní architekturu PLC a jeho procesor, což by v ostatních programovacích jazycích mohlo způsobit nemalé problémy.

Použitá literatura

- [1] Rydlo, P.: *Krokové motory a jejich řízení*. TU Liberec 2000
- [2] *Krokové motory* [online]
URL: <<http://educon.zcu.cz/view.php?cislomodulu=2006021101>>
- [3] Řezáč, K.: *Krokové motory* [online]
URL: <<http://robotika.cz/articles/steppers/cs>>
- [4] *Pohony s krokovými motorky* [online]
URL: <<http://fei1.vsb.cz/kat448/Studium/Materialy/VS/kap26.pdf>>
- [5] *Lineární motory* [online]
URL: <<http://www.vues.eu/doc/?docid=73>>
- [6] *Linární pohony* [online]
URL: <http://www.hiwin.cz/pdf/clanek_o_linearnich_motorech_-_technika_a_trh.pdf>
- [7] Novotný, J. - Hnízdil, J.: *Rotační a lineární krokové motory* [online]
URL: <<http://www.elektrorevue.cz/clanky/05043/>>
- [8] *Bezželezné lineární motory* [online]
URL: <<http://www.odbornecasopisy.cz/automa/2006/au010625.htm>>
- [9] Noskijevič, P. - Jániš, P.: *Simulace a řízení pneumatického servopohonu pomocí programu matlab – simulink* [online]
URL: <http://dsp.vscht.cz/konference_matlab/matlab04/noskievic.pdf>
- [10] *Hnací impulsy elektrických a pneumatických válců* [online]
URL: <http://www.festo.com/INetDomino/innovations_2006/cs/bdda62e9b7ba00bac125713e0049423e.htm>
- [11] *PROFI PNEUMATICS II* [online]
URL: <http://www.stavebnice.com/eshop/images/manual_profi_pneumatic2cz.pdf>
- [12] *Manipulátor s proporcionálními pneumatickými pohony* [online]
URL: <www.ksk.vslib.cz/podklady/ZR-FS/Cviceni11.pdf>
- [13] Mitsubishi Electric. *Melservo. MR-MG30. Profibus/DP Option Unit. Servo Amplifiers and Motors. Instruction Manual*. Art. no.: 159569. 15.06.2004
- [14] *PROFIBUS Technology and Application. System Description*. Open Solutions

- for the World Automation. Version October 2002. Order Number 4.002
- [15] *PROFIBUS Specification*. European Standard EN 50 170 Volume 2. Edition 1.0. March 1998
- [16] *PROFIBUS Specification PROFIdrive – Profil, Drive Technology*. Version 3.1.2 September 2004, Order No: 3.172
- [17] Mitsubishi Electric. *GX IEC Developer 7.00, Reference Manual*. IEC Programing and Documentation System. Art. no.: 43597. 22.09.2005.
- [18] Mitsubishi Electric. *MELSERVO – J2-Super Series, SSCNET Compatible, Model MR-J2S- B. SERVO AMPLIFIER INSTRUCTION MANUAL*.
- [19] *Modular PLC – MELSEC System Q* [online]
URL: <http://www.mitsubishi-automation.co.uk/products/modularplc_SYSTEMQ.html>
- [20] Mitsubishi Electric. *E-Designer, Operation manual*. Programing software for Graphic Operator Terminals E300, E600, E610, E615, E700, E710, E900, E900VT and E910. Art. no.: 149968. 09.11.2004.

Příloha A

DA	Adresa cíle (Destination Address)
Data Unit	Datové pole: pevná délka: $L - 3 = 8$ oktetů \rightarrow rámec s pevnou délkou dat proměnná délka $L - 3 =$ maximálně 246 oktetů \rightarrow rámec s proměnnou délkou dat
ED	Koncový znak (End Delimiter), hodnota: 16H \rightarrow pro všechny typy rámců
FC	Řídící pole (Frame Control)
FCS	Kontrolní součet (Frame Check Sequence)
L	Délka informace (Information Field Length): počet oktetů: $L = 3 \rightarrow$ rámec bez dat počet oktetů: $L = 11 \rightarrow$ rámec s pevnou délkou dat počet oktetů : $L = 4 - 249 \rightarrow$ rámec s proměnnou délkou dat
LE	Délka oktetu (Octet Length)
Ler	Délka oktetu opakovaně (Octet Length repeated)
SA	Adresa zdroje (Source Address)
SC	Jeden znak (Single Character), hodnota: E5H
SD1	Startovací znak (Start Delimiter), hodnota: 10H \rightarrow rámec bez dat
SD2	Startovací znak (Start Delimiter), hodnota: 68H \rightarrow rámec s proměnnou délkou dat
SD3	Startovací znak (Start Delimiter), hodnota: A2H \rightarrow rámec s pevnou délkou dat
SD4	Startovací znak (Start Delimiter), hodnota: DCH \rightarrow rámec pověření

Příloha B

Význam jednotlivých označení slov v datovém rámci PPO typu 5:

Rozdělení PKW:

IND	Index pole
PKE	Charakteristika (identifikace) parametru
PWE	Hodnota parametru

Rozdělení PZD:

HIW	Hlavní skutečná hodnota
HSW	Hlavní zadaná hodnota
PZD1 – PZD10	Data daného procesu (určitá aplikace)
STW1	Řídící slovo
ZSW1	Stavové slovo

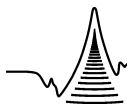
Význam jednotlivých označení slov v datovém rámci, jenž vysílá jednotka MR-MG30:

Rozdělení PKW:

PKE	Charakteristika parametru
IND	Sub-index pole
PWE	Čtení a zápis jednotlivých požadavků

Rozdělení PZD:

ACCEL	Akcelerační časová konstanta (požadavek na příkaz)
APOSH	Horních 16 bitů aktuální obecné polohy (odezva na příkaz)
APOSL	Spodních 16 bitů aktuální obecné polohy (odezva na příkaz)
ASPD	Rychlost motoru (odezva na příkaz)
AUX	Pomocná funkce (požadavek na příkaz)
CMD1	Řídící příkaz 1 (požadavek na příkaz)
CMD2	Řídící příkaz 2 (požadavek na příkaz)
CPOSH	Horních 16 bitů polohy – souřadnic (požadavek na příkaz)



CPOSL	Spodních 16 bitů polohy – souřadnic (požadavek na příkaz)
CSPD	Rychlost příkazu – rychlost vykonávání (požadavek na příkaz)
DECEL	Deakcelerační časová konstanta (požadavek na příkaz)
Reserved	Vyhrazeno pro rozšíření (expanzi)
STS1	Stav ovladače 1 (odezva na příkaz)
STS2	Stav ovladače 2 (odezva na příkaz)
STW	Řídící (kontrolní) slovo (požadavek na příkaz)
TRQ	Okamžitý točivý moment (odezva na příkaz)
ZSW	Stavové slovo (odezva na příkaz)

Obsazení jednotlivých bitů PKW:

AK	Požadavek na příkaz (vstupní data), nebo odezva na příkaz (výstupní data)
PNU	Číslo parametru
PWE1	Horních 16 bitů (zápis nebo čtení hodnoty parametru)
PWE2	Spodních 16 bitů (zápis nebo čtení hodnoty parametru)
Reserved	Vyhrazeno pro rozšíření (expanzi)
SPM	Požadavek na příkaz (provedení úlohy požadavku), nebo odezva na příkaz (prováděná úloha je dokončena)
Sub-Index	Pod-Index (číslo pole)

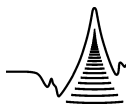
Obsazení jednotlivých bitů PZD pro výstupní data (požadavek na příkaz):

ACCEL	Časová konstanta zrychlení
AUX	Pomocná funkce
CMD2	Pro nastavení výrobcem
CPOSH	Horních 16 bitů polohy (souřadnic)
CPOSL	Dolních 16 bitů polohy (souřadnic)
CSPD	Rychlost servomotoru
DECEL	Časová konstanta zpomalení
DIR	Start řízení

DOG	Blízkost dorazu
ITP	Přerušení polohování
LSN	Krajní poloha dopředné rotace
LSP	Krajní poloha zpětné rotace
MD0	Pracovní režim 0
MD1	Pracovní režim 1
MD2	Pracovní režim 2
ORST	Reset alarmu činnosti
PIF	Příznak počátečního vysílání
RST	Zrušení alarmu servomotoru
SCHG	Změna rychlosti
SON	Příznak napájení servomotoru
ST	Start provozu
STP	Zastavení provozu
TL	Výběr omezení točivého momentu
TSTP	Dočasné zastavení
X	Vyhrazeno pro nastavení výrobcem

Obsazení jednotlivých bitů PZD pro vstupní data (odezva na příkaz):

APOSH	Horních 16 bitů aktuální polohy
APOSL	Dolních 16 bitů aktuální polohy
ASPD	Rychlost servomotoru
CPO	Hrubá shoda polohy
CR	Řídící požadavek
Fault	Hlášení o alarmu servomotoru nebo jednotky
INP	Příznak, zda došlo k poklesu pulsů pod nastavenou hodnotu
MD0	Výstupní pracovní režim 0
MD1	Výstupní pracovní režim 1
MD2	Výstupní pracovní režim 2
MEND	Dokončení pohybu
OALM	Alarm provozu



POP	Zákaz napájení
POT	Výstupní rozsah polohování
PUS	Dočasné zastavení
RDONS	Příznak dokončení inicializace servomotoru
SCE	Neplatná změna rychlosti
SCF	Korektně dokončená změna rychlosti
SONS	Příznak možnosti spuštění některého z provozních režimů
STS2	Pro nastavení výrobcem
TLC	Omezení točivého momentu
TRQ	Okamžitý točivý moment
WNG	Hlášení (warning) servomotoru
X	Vyhrazeno pro nastavení výrobcem
ZP	Příznak dosažení nulové polohy

Příloha C

Globální proměnná	Popis
AlarmServoOrMR_MG30Ax1 (2)	Alarm servomotoru nebo modulu Profibus
AlarmWarningCodeAx1 (2)	Kód alarmu, který nastal v servomotoru nebo v MR-MG30
CurrentPositionAx1 (2)	Aktuální poloha servomotoru
CurrentSpeedAx1 (2)	Aktuální rychlost servomotoru
CurrentTorqueAx1 (2)	Aktuální točivý moment servomotoru
FromServoAx1 (2)	Posloupnost čtrnácti řídicích datových slov, která jsou přijímána od servoměniče
HomeCompletedAx1 (2)	Vrácení nulové polohy dokončeno
IndexAx1 (2)	Indexace parametru (servomotoru)
Next_IndAx1 (2)	Start polohování příslušné osy do následující polohy
NextPosition	Start polohování servopohonu do následující polohy
OperationAlarmAx1 (2)	Indikace provozní chyby
OperationErrorCodeAx1 (2)	Kód provozní chyby, která nastala
PNUNoAx1 (2)	Číslo parametru
PNUValueAx1 (2)	Hodnota parametru
PoAx1 (2)	Pořadové číslo polohy, ve které se nachází příslušná osa servopohonu
Pos_UnloadAx1 (2)	Polohy příslušné osy byly načteny
PosCompletedAx1 (2)	Polohování dokončeno
ReadPNUAx1 (2)	Čtení požadovaného parametru
ResetAlarmAx1 (2)	Reset alarmu, pokud nastal
ServoInitializedAx1 (2)	Proběhla inicializace komunikace se servosystémem
ServoInPositionAx1 (2)	Servomotor je v požadované pozici
ServoOnAx1 (2)	Zapnutí / vypnutí servomotoru
ServoReadyAx1 (2)	Servomotor je připraven k práci

StartPosAx1 (2)	Zahájení polohování příslušné osy servopohonu
StartPositioning	Zahájení polohování servopohonu
ToServoAx1 (2)	Posloupnost čtrnácti řídicích datových slov, která jsou posílána do servoměniče
WarningServoAx1 (2)	Hlášení (warning) servomotoru
WritePNUAx1 (2)	Zápis požadovaného parametru